UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1996

HIGH SCHOOL APPRENTICESHIP PROGRAM FINAL REPORTS

VOLUME 14

ROME LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major Linda Steel-Goodwin

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Johnetta Thompson

Program Administrator, RDL
Rebecca Kelly

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1996

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

This document is one of a set of 16 volumes describing the 1996 AFOSR Summer Research Program. The following volumes comprise the set:

VOLUME	TITLE
1	Program Management Report
	Summer Faculty Research Program (SFRP) Reports
2A & 2B	Armstrong Laboratory
3A & 3B	Phillips Laboratory
4	Rome Laboratory
5A, 5B & 5C	Wright Laboratory
6	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	Graduate Student Research Program (GSRP) Reports
7A & 7B	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10A & 10B	Wright Laboratory
11	Arnold Engineering Development Center, United States Air Force Academy, Wilford Hall Medical Center and Wright Patterson Medical Center
	High School Apprenticeship Program (HSAP) Reports
12A & 12B	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A&15B	Wright Laboratory
16	Arnold Engineering Development Center

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

INTRODUCTION

The Summer Research Program (SRP), sponsored by the Air Force Office of Scientific Research (AFOSR), offers paid opportunities for university faculty, graduate students, and high school students to conduct research in U.S. Air Force research laboratories nationwide during the summer.

Introduced by AFOSR in 1978, this innovative program is based on the concept of teaming academic researchers with Air Force scientists in the same disciplines using laboratory facilities and equipment not often available at associates' institutions.

The Summer Faculty Research Program (SFRP) is open annually to approximately 150 faculty members with at least two years of teaching and/or research experience in accredited U.S. colleges, universities, or technical institutions. SFRP associates must be either U.S. citizens or permanent residents.

The Graduate Student Research Program (GSRP) is open annually to approximately 100 graduate students holding a bachelor's or a master's degree; GSRP associates must be U.S. citizens enrolled full time at an accredited institution.

The High School Apprentice Program (HSAP) annually selects about 125 high school students located within a twenty mile commuting distance of participating Air Force laboratories.

AFOSR also offers its research associates an opportunity, under the Summer Research Extension Program (SREP), to continue their AFOSR-sponsored research at their home institutions through the award of research grants. In 1994 the maximum amount of each grant was increased from $20,000 to $25,000, and the number of AFOSR-sponsored grants decreased from 75 to 60. A separate annual report is compiled on the SREP.

The numbers of projected summer research participants in each of the three categories and SREP "grants" are usually increased through direct sponsorship by participating laboratories.

AFOSR's SRP has well served its objectives of building critical links between Air Force research laboratories and the academic community, opening avenues of communications and forging new research relationships between Air Force and academic technical experts in areas of national interest, and strengthening the nation's efforts to sustain careers in science and engineering. The success of the SRP can be gauged from its growth from inception (see Table 1) and from the favorable responses the 1996 participants expressed in end-of-tour SRP evaluations (Appendix B).

AFOSR contracts for administration of the SRP by civilian contractors. The contract was first awarded to Research & Development Laboratories (RDL) in September 1990. After

completion of the 1990 contract, RDL (in 1993) won the recompetition for the basic year and four 1-year options.

2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM

The SRP began with faculty associates in 1979; graduate students were added in 1982 and high school students in 1986. The following table shows the number of associates in the program each year.

YEAR	SRP Participation, by Year			TOTAL
	SFRP	GSRP	HSAP	
1979	70			70
1980	87			87
1981	87			87
1982	91	17		108
1983	101	53		154
1984	152	84		236
1985	154	92		246
1986	158	100	42	300
1987	159	101	73	333
1988	153	107	101	361
1989	168	102	103	373
1990	165	121	132	418
1991	170	142	132	444
1992	185	121	159	464
1993	187	117	136	440
1994	192	117	133	442
1995	190	115	137	442
1996	188	109	138	435

Beginning in 1993, due to budget cuts, some of the laboratories weren't able to afford to fund as many associates as in previous years. Since then, the number of funded positions has remained fairly constant at a slightly lower level.

3. RECRUITING AND SELECTION

The SRP is conducted on a nationally advertised and competitive-selection basis. The advertising for faculty and graduate students consisted primarily of the mailing of 8,000 52-page SRP brochures to chairpersons of departments relevant to AFOSR research and to administrators of grants in accredited universities, colleges, and technical institutions. Historically Black Colleges and Universities (HBCUs) and Minority Institutions (MIs) were included. Brochures also went to all participating USAF laboratories, the previous year's participants, and numerous individual requesters (over 1000 annually).

RDL placed advertisements in the following publications: *Black Issues in Higher Education, Winds of Change,* and *IEEE Spectrum.* Because no participants list either *Physics Today* or *Chemical & Engineering News* as being their source of learning about the program for the past several years, advertisements in these magazines were dropped, and the funds were used to cover increases in brochure printing costs.

High school applicants can participate only in laboratories located no more than 20 miles from their residence. Tailored brochures on the HSAP were sent to the head counselors of 180 high schools in the vicinity of participating laboratories, with instructions for publicizing the program in their schools. High school students selected to serve at Wright Laboratory's Armament Directorate (Eglin Air Force Base, Florida) serve eleven weeks as opposed to the eight weeks normally worked by high school students at all other participating laboratories.

Each SFRP or GSRP applicant is given a first, second, and third choice of laboratory. High school students who have more than one laboratory or directorate near their homes are also given first, second, and third choices.

Laboratories make their selections and prioritize their nominees. AFOSR then determines the number to be funded at each laboratory and approves laboratories' selections.

Subsequently, laboratories use their own funds to sponsor additional candidates. Some selectees do not accept the appointment, so alternate candidates are chosen. This multi-step selection procedure results in some candidates being notified of their acceptance after scheduled deadlines. The total applicants and participants for 1996 are shown in this table.

1996 Applicants and Participants			
PARTICIPANT CATEGORY	TOTAL APPLICANTS	SELECTEES	DECLINING SELECTEES
SFRP	572	188	39
(HBCU/MI)	(119)	(27)	(5)
GSRP	235	109	7
(HBCU/MI)	(18)	(7)	(1)
HSAP	474	138	8
TOTAL	1281	435	54

4. SITE VISITS

During June and July of 1996, representatives of both AFOSR/NI and RDL visited each participating laboratory to provide briefings, answer questions, and resolve problems for both laboratory personnel and participants. The objective was to ensure that the SRP would be as constructive as possible for all participants. Both SRP participants and RDL representatives found these visits beneficial. At many of the laboratories, this was the only opportunity for all participants to meet at one time to share their experiences and exchange ideas.

5. HISTORICALLY BLACK COLLEGES AND UNIVERSITIES AND MINORITY INSTITUTIONS (HBCU/MIs)

Before 1993, an RDL program representative visited from seven to ten different HBCU/Mis annually to promote interest in the SRP among the faculty and graduate students. These efforts were marginally effective, yielding a doubling of HBCI/MI applicants. In an effort to achieve AFOSR's goal of 10% of all applicants and selectees being HBCU/MI qualified, the RDL team decided to try other avenues of approach to increase the number of qualified applicants. Through the combined efforts of the AFOSR Program Office at Bolling AFB and RDL, two very active minority groups were found, HACU (Hispanic American Colleges and Universities) and AISES (American Indian Science and Engineering Society). RDL is in communication with representatives of each of these organizations on a monthly basis to keep up with the their activities and special events. Both organizations have widely-distributed magazines/quarterlies in which RDL placed ads.

Since 1994 the number of both SFRP and GSRP HBCU/MI applicants and participants has increased ten-fold, from about two dozen SFRP applicants and a half dozen selectees to over 100 applicants and two dozen selectees, and a half-dozen GSRP applicants and two or three selectees to 18 applicants and 7 or 8 selectees. Since 1993, the SFRP had a two-fold applicant

increase and a two-fold selectee increase. Since 1993, the GSRP had a three-fold applicant increase and a three to four-fold increase in selectees.

In addition to RDL's special recruiting efforts, AFOSR attempts each year to obtain additional funding or use leftover funding from cancellations the past year to fund HBCU/MI associates. This year, 5 HBCU/MI SFRPs declined after they were selected (and there was no one qualified to replace them with). The following table records HBCU/MI participation in this program.

SRP HBCU/MI Participation, By Year				
YEAR	SFRP		GSRP	
	Applicants	Participants	Applicants	Participants
1985	76	23	15	11
1986	70	18	20	10
1987	82	32	32	10
1988	53	17	23	14
1989	39	15	13	4
1990	43	14	17	3
1991	42	13	8	5
1992	70	13	9	5
1993	60	13	6	2
1994	90	16	11	6
1995	90	21	20	8
1996	119	27	18	7

6. SRP FUNDING SOURCES

Funding sources for the 1996 SRP were the AFOSR-provided slots for the basic contract and laboratory funds. Funding sources by category for the 1996 SRP selected participants are shown here.

1996 SRP FUNDING CATEGORY	SFRP	GSRP	HSAP
AFOSR Basic Allocation Funds	141	85	123
USAF Laboratory Funds	37	19	15
HBCU/MI By AFOSR (Using Procured Addn'l Funds)	10	5	0
TOTAL	188	109	138

SFRP - 150 were selected, but nine canceled too late to be replaced.
GSRP - 90 were selected, but five canceled too late to be replaced (10 allocations for the ALCs were withheld by AFOSR.)
HSAP - 125 were selected, but two canceled too late to be replaced.

7. COMPENSATION FOR PARTICIPANTS

Compensation for SRP participants, per five-day work week, is shown in this table.

1996 SRP Associate Compensation

PARTICIPANT CATEGORY	1991	1992	1993	1994	1995	1996
Faculty Members	$690	$718	$740	$740	$740	$770
Graduate Student (Master's Degree)	$425	$442	$455	$455	$455	$470
Graduate Student (Bachelor's Degree)	$365	$380	$391	$391	$391	$400
High School Student (First Year)	$200	$200	$200	$200	$200	$200
High School Student (Subsequent Years)	$240	$240	$240	$240	$240	$240

The program also offered associates whose homes were more than 50 miles from the laboratory an expense allowance (seven days per week) of $50/day for faculty and $40/day for graduate students. Transportation to the laboratory at the beginning of their tour and back to their home destinations at the end was also reimbursed for these participants. Of the combined SFRP and

GSRP associates, 65 % (194 out of 297) claimed travel reimbursements at an average round-trip cost of $780.

Faculty members were encouraged to visit their laboratories before their summer tour began. All costs of these orientation visits were reimbursed. Forty-five percent (85 out of 188) of faculty associates took orientation trips at an average cost of $444. By contrast, in 1993, 58 % of SFRP associates took orientation visits at an average cost of $685; that was the highest percentage of associates opting to take an orientation trip since RDL has administered the SRP, and the highest average cost of an orientation trip. These 1993 numbers are included to show the fluctuation which can occur in these numbers for planning purposes.

Program participants submitted biweekly vouchers countersigned by their laboratory research focal point, and RDL issued paychecks so as to arrive in associates' hands two weeks later.

In 1996, RDL implemented direct deposit as a payment option for SFRP and GSRP associates. There were some growing pains. Of the 128 associates who opted for direct deposit, 17 did not check to ensure that their financial institutions could support direct deposit (and they couldn't), and eight associates never did provide RDL with their banks' ABA number (direct deposit bank routing number), so only 103 associates actually participated in the direct deposit program. The remaining associates received their stipend and expense payments via checks sent in the US mail.

HSAP program participants were considered actual RDL employees, and their respective state and federal income tax and Social Security were withheld from their paychecks. By the nature of their independent research, SFRP and GSRP program participants were considered to be consultants or independent contractors. As such, SFRP and GSRP associates were responsible for their own income taxes, Social Security, and insurance.

8. CONTENTS OF THE 1996 REPORT

The complete set of reports for the 1996 SRP includes this program management report (Volume 1) augmented by fifteen volumes of final research reports by the 1996 associates, as indicated below:

1996 SRP Final Report Volume Assignments

LABORATORY	SFRP	GSRP	HSAP
Armstrong	2	7	12
Phillips	3	8	13
Rome	4	9	14
Wright	5A, 5B	10	15
AEDC, ALCs, WHMC	6	11	16

APPENDIX A – PROGRAM STATISTICAL SUMMARY

A. Colleges/Universities Represented

Selected SFRP associates represented 169 different colleges, universities, and institutions, GSRP associates represented 95 different colleges, universities, and institutions.

B. States Represented

SFRP -Applicants came from 47 states plus Washington D.C. and Puerto Rico. Selectees represent 44 states plus Puerto Rico.

GSRP - Applicants came from 44 states and Puerto Rico. Selectees represent 32 states.

HSAP - Applicants came from thirteen states. Selectees represent nine states.

Total Number of Participants	
SFRP	188
GSRP	109
HSAP	138
TOTAL	435

Degrees Represented			
	SFRP	GSRP	TOTAL
Doctoral	184	1	185
Master's	4	48	52
Bachelor's	0	60	60
TOTAL	188	109	297

SFRP Academic Titles	
Assistant Professor	79
Associate Professor	59
Professor	42
Instructor	3
Chairman	0
Visiting Professor	1
Visiting Assoc. Prof.	0
Research Associate	4
TOTAL	188

Source of Learning About the SRP		
Category	Applicants	Selectees
Applied/participated in prior years	28%	34%
Colleague familiar with SRP	19%	16%
Brochure mailed to institution	23%	17%
Contact with Air Force laboratory	17%	23%
IEEE Spectrum	2%	1%
BIIHE	1%	1%
Other source	10%	8%
TOTAL	100%	100%

APPENDIX B – SRP EVALUATION RESPONSES

1. OVERVIEW

Evaluations were completed and returned to RDL by four groups at the completion of the SRP. The number of respondents in each group is shown below.

Table B-1. Total SRP Evaluations Received

Evaluation Group	Responses
SFRP & GSRPs	275
HSAPs	113
USAF Laboratory Focal Points	84
USAF Laboratory HSAP Mentors	6

All groups indicate unanimous enthusiasm for the SRP experience.

The summarized recommendations for program improvement from both associates and laboratory personnel are listed below:

A. Better preparation on the labs' part prior to associates' arrival (i.e., office space, computer assets, clearly defined scope of work).

B. Faculty Associates suggest higher stipends for SFRP associates.

C. Both HSAP Air Force laboratory mentors and associates would like the summer tour extended from the current 8 weeks to either 10 or 11 weeks; the groups state it takes 4-6 weeks just to get high school students up-to-speed on what's going on at laboratory. (Note: this same argument was used to raise the faculty and graduate student participation time a few years ago.)

2. 1996 USAF LABORATORY FOCAL POINT (LFP) EVALUATION RESPONSES

The summarized results listed below are from the 84 LFP evaluations received.

1. LFP evaluations received and associate preferences:

Table B-2. Air Force LFP Evaluation Responses (By Type)

		How Many Associates Would You Prefer To Get ? (% Response)											
		SFRP				GSRP (w/Univ Professor)				GSRP (w/o Univ Professor)			
Lab	Evals Recv'd	0	1	2	3+	0	1	2	3+	0	1	2	3+
AEDC	0	-	-	-	-	-	-	-	-	-	-	-	-
WHMC	0	-	-	-	-	-	-	-	-	-	-	-	-
AL	7	28	28	28	14	54	14	28	0	86	0	14	0
FJSRL	1	0	100	0	0	100	0	0	0	0	100	0	0
PL	25	40	40	16	4	88	12	0	0	84	12	4	0
RL	5	60	40	0	0	80	10	0	0	100	0	0	0
WL	46	30	43	20	6	78	17	4	0	93	4	2	0
Total	84	32%	50%	13%	5%	80%	11%	6%	0%	73%	23%	4%	0%

LFP Evaluation Summary. The summarized responses, by laboratory, are listed on the following page. LFPs were asked to rate the following questions on a scale from 1 (below average) to 5 (above average).

2. LFPs involved in SRP associate application evaluation process:
    a. Time available for evaluation of applications:
    b. Adequacy of applications for selection process:
3. Value of orientation trips:
4. Length of research tour:
5. a. Benefits of associate's work to laboratory:
    b. Benefits of associate's work to Air Force:
6. a. Enhancement of research qualifications for LFP and staff:
    b. Enhancement of research qualifications for SFRP associate:
    c. Enhancement of research qualifications for GSRP associate:
7. a. Enhancement of knowledge for LFP and staff:
    b. Enhancement of knowledge for SFRP associate:
    c. Enhancement of knowledge for GSRP associate:
8. Value of Air Force and university links:
9. Potential for future collaboration:
10. a. Your working relationship with SFRP:
    b. Your working relationship with GSRP:
11. Expenditure of your time worthwhile:

(Continued on next page)

12. Quality of program literature for associate:
13. a. Quality of RDL's communications with you:
    b. Quality of RDL's communications with associates:
14. Overall assessment of SRP:

Table B-3. Laboratory Focal Point Reponses to above questions

	AEDC	*AL*	*FJSRL*	*PL*	*RL*	*WHMC*	*WL*
# Evals Recv'd	0	7	1	14	5	0	46
Question #							
2	-	86 %	0 %	88 %	80 %	-	85 %
2a	-	4.3	n/a	3.8	4.0	-	3.6
2b	-	4.0	n/a	3.9	4.5	-	4.1
3	-	4.5	n/a	4.3	4.3	-	3.7
4	-	4.1	4.0	4.1	4.2	-	3.9
5a	-	4.3	5.0	4.3	4.6	-	4.4
5b	-	4.5	n/a	4.2	4.6	-	4.3
6a	-	4.5	5.0	4.0	4.4	-	4.3
6b	-	4.3	n/a	4.1	5.0	-	4.4
6c	-	3.7	5.0	3.5	5.0	-	4.3
7a	-	4.7	5.0	4.0	4.4	-	4.3
7b	-	4.3	n/a	4.2	5.0	-	4.4
7c	-	4.0	5.0	3.9	5.0	-	4.3
8	-	4.6	4.0	4.5	4.6	-	4.3
9	-	4.9	5.0	4.4	4.8	-	4.2
10a	-	5.0	n/a	4.6	4.6	-	4.6
10b	-	4.7	5.0	3.9	5.0	-	4.4
11	-	4.6	5.0	4.4	4.8	-	4.4
12	-	4.0	4.0	4.0	4.2	-	3.8
13a	-	3.2	4.0	3.5	3.8	-	3.4
13b	-	3.4	4.0	3.6	4.5	-	3.6
14	-	4.4	5.0	4.4	4.8	-	4.4

3. 1996 SFRP & GSRP EVALUATION RESPONSES

The summarized results listed below are from the 257 SFRP/GSRP evaluations received.

Associates were asked to rate the following questions on a scale from 1 (below average) to 5 (above average) - by Air Force base results and over-all results of the 1996 evaluations are listed after the questions.

1. The match between the laboratories research and your field:
2. Your working relationship with your LFP:
3. Enhancement of your academic qualifications:
4. Enhancement of your research qualifications:
5. Lab readiness for you: LFP, task, plan:
6. Lab readiness for you: equipment, supplies, facilities:
7. Lab resources:
8. Lab research and administrative support:
9. Adequacy of brochure and associate handbook:
10. RDL communications with you:
11. Overall payment procedures:
12. Overall assessment of the SRP:
13. a. Would you apply again?
    b. Will you continue this or related research?
14. Was length of your tour satisfactory?
15. Percentage of associates who experienced difficulties in finding housing:
16. Where did you stay during your SRP tour?
    a. At Home:
    b. With Friend:
    c. On Local Economy:
    d. Base Quarters:
17. Value of orientation visit:
    a. Essential:
    b. Convenient:
    c. Not Worth Cost:
    d. Not Used:

SFRP and GSRP associate's responses are listed in tabular format on the following page.

Table B-4. 1996 SFRP & GSRP Associate Responses to SRP Evaluation

	Arnold	Brooks	Edwards	Eglin	Griffis	Hanscom	Kelly	Kirtland	Lackland	Robins	Tyndall	WPAFB	average
# res	6	48	6	14	31	19	3	32	1	2	10	85	257
1	4.8	4.4	4.6	4.7	4.4	4.9	4.6	4.6	5.0	5.0	4.0	4.7	4.6
2	5.0	4.6	4.1	4.9	4.7	4.7	5.0	4.7	5.0	5.0	4.6	4.8	4.7
3	4.5	4.4	4.0	4.6	4.3	4.2	4.3	4.4	5.0	5.0	4.5	4.3	4.4
4	4.3	4.5	3.8	4.6	4.4	4.4	4.3	4.6	5.0	4.0	4.4	4.5	4.5
5	4.5	4.3	3.3	4.8	4.4	4.5	4.3	4.2	5.0	5.0	3.9	4.4	4.4
6	4.3	4.3	3.7	4.7	4.4	4.5	4.0	3.8	5.0	5.0	3.8	4.2	4.2
7	4.5	4.4	4.2	4.8	4.5	4.3	4.3	4.1	5.0	5.0	4.3	4.3	4.4
8	4.5	4.6	3.0	4.9	4.4	4.3	4.3	4.5	5.0	5.0	4.7	4.5	4.5
9	4.7	4.5	4.7	4.5	4.3	4.5	4.7	4.3	5.0	5.0	4.1	4.5	4.5
10	4.2	4.4	4.7	4.4	4.1	4.1	4.0	4.2	5.0	4.5	3.6	4.4	4.3
11	3.8	4.1	4.5	4.0	3.9	4.1	4.0	4.0	3.0	4.0	3.7	4.0	4.0
12	5.7	4.7	4.3	4.9	4.5	4.9	4.7	4.6	5.0	4.5	4.6	4.5	4.6
Numbers below are percentages													
13a	83	90	83	93	87	75	100	81	100	100	100	86	87
13b	100	89	83	100	94	98	100	94	100	100	100	94	93
14	83	96	100	90	87	80	100	92	100	100	70	84	88
15	17	6	0	33	20	76	33	25	0	100	20	8	39
16a	-	26	17	9	38	23	33	4	-	-	-	30	
16b	100	33	-	40	-	8	-	-	-	-	36	2	
16c	-	41	83	40	62	69	67	96	100	100	64	68	
16d	-	-	-	-	-	-	-	-	-	-	-	0	
17a	-	33	100	17	50	14	67	39	-	50	40	31	35
17b	-	21	-	17	10	14	-	24	-	50	20	16	16
17c	-	-	-	-	10	7	-	-	-	-	-	2	3
17d	100	46	-	66	30	69	33	37	100	-	40	51	46

4. 1996 USAF LABORATORY HSAP MENTOR EVALUATION RESPONSES

Not enough evaluations received (5 total) from Mentors to do useful summary.

5. 1996 HSAP EVALUATION RESPONSES

The summarized results listed below are from the 113 HSAP evaluations received.

HSAP apprentices were asked to rate the following questions on a scale from 1 (below average) to 5 (above average)

1. Your influence on selection of topic/type of work.
2. Working relationship with mentor, other lab scientists.
3. Enhancement of your academic qualifications.
4. Technically challenging work.
5. Lab readiness for you: mentor, task, work plan, equipment.
6. Influence on your career.
7. Increased interest in math/science.
8. Lab research & administrative support.
9. Adequacy of RDL's Apprentice Handbook and administrative materials.
10. Responsiveness of RDL communications.
11. Overall payment procedures.
12. Overall assessment of SRP value to you.
13. Would you apply again next year? Yes (92 %)
14. Will you pursue future studies related to this research? Yes (68 %)
15. Was Tour length satisfactory? Yes (82 %)

	Arnold	Brooks	Edwards	Eglin	Griffiss	Hanscom	Kirtland	Tyndall	WPAFB	Totals
# resp	5	19	7	15	13	2	7	5	40	113
1	2.8	3.3	3.4	3.5	3.4	4.0	3.2	3.6	3.6	3.4
2	4.4	4.6	4.5	4.8	4.6	4.0	4.4	4.0	4.6	4.6
3	4.0	4.2	4.1	4.3	4.5	5.0	4.3	4.6	4.4	4.4
4	3.6	3.9	4.0	4.5	4.2	5.0	4.6	3.8	4.3	4.2
5	4.4	4.1	3.7	4.5	4.1	3.0	3.9	3.6	3.9	4.0
6	3.2	3.6	3.6	4.1	3.8	5.0	3.3	3.8	3.6	3.7
7	2.8	4.1	4.0	3.9	3.9	5.0	3.6	4.0	4.0	3.9
8	3.8	4.1	4.0	4.3	4.0	4.0	4.3	3.8	4.3	4.2
9	4.4	3.6	4.1	4.1	3.5	4.0	3.9	4.0	3.7	3.8
10	4.0	3.8	4.1	3.7	4.1	4.0	3.9	2.4	3.8	3.8
11	4.2	4.2	3.7	3.9	3.8	3.0	3.7	2.6	3.7	3.8
12	4.0	4.5	4.9	4.6	4.6	5.0	4.6	4.2	4.3	4.5
Numbers below are percentages										
13	60%	95%	100%	100%	85%	100%	100%	100%	90%	92%
14	20%	80%	71%	80%	54%	100%	71%	80%	65%	68%
15	100%	70%	71%	100%	100%	50%	86%	60%	80%	82%

Robert C. Altshuler's report was not available at the time of publication.

Michael A. Bartley's report was not available at the time of publication.

PREPARATION AND PLACEMENT OF MATERIALS ON THE WORLD-WIDE WEB

Daniel T. Brown

Sauquoit Valley High School
Oneida Street
Sauquoit, NY 13456

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

August 1996

PREPARATION AND PLACEMENT OF MATERIALS ON THE WORLD-WIDE WEB

Daniel T. Brown
Sauquoit Valley High School

Abstract

The website of the Intelligence and Reconnaissance Directorate, Image Products Division (IRR) was expanded by adding documents and new in-lined images to provide more information to the public and to allow people outside Rome Laboratory to monitor the progress on non-classified projects. This was accomplished through the use of HTML 2.0, Adobe Photoshop 3.0, Microsoft Word 6.0.1, GraphicConverter, SimpleText, and digital pictures handled with a SCSII interface. The HTML documents were written into Microsoft Word and SimpleText documents by adding in the HTML tags and saving as a more accepted Text Only format. The digital pictures were edited and layered using Photoshop to achieve a graphics format that is accepted by GraphicConverter. The next step was to convert the PICT files into a format accepted by a Netscape browser (JPEG format). The documents and graphics were then passed on to the system administrator for placement on the IRR server as to be accessible on the World-Wide Web.

PREPARATION AND PLACEMENT OF MATERIALS ON THE WORLD-WIDE WEB

Daniel T. Brown
Sauquoit Valley High School

Introduction

In today's ever-changing world, the Internet or World-Wide Web (WWW) has gained enormous amounts of power. Businesses are able to negotiate across several thousand miles with little to no delay between messages. This is the advantage that Rome Laboratory's Image Products Division seeks. This is why it maintains a website--to attract business through the display of its research interests, ongoing research, and past successes. By making these accomplishments known to the public, more contracts can be attracted to Rome Laboratory, drawn by interest in its past projects and the off-shoot technologies produced there. However, in the Information Age, speed is of the essence. Pages on the Internet must be updated, expanded, and extra features must be added. Failure to keep up can result in catastrophic results, for people today will seek the most recent information available. Individual homepages were added to the roster, so the Image Products Division, Image Exploitation Branch members could be contacted and their ongoing research discussed. Point papers were linked so project progress could also be observed and followed as the pages are updated in the future. This major addition was something that will hopefully spark interest around the world concerning what the Image Products Division, Image Exploitation Branch (RL/IRRE) can do for them.

Methodology

The additions to the Rome Laboratory Image Exploitation Branch's website were created using HTML (HytperText Mark-up Language) 2.0. This was the result of wishes to keep the page accessible by the greatest amount of people including those with text-only browsers. Java and JavaScript were considered, but rejected when both were found to compromise system integrity and security. By downloading the source document of an existing employee homepage, and editing out the specific information, a template that allows one to create the HTML documents was designed.

RL/IRRE Personal Homepage Template

```
<HTML>
<HEAD>
<TITLE>
<!--NAME-->
```

```
</TITLE>
</HEAD>
<BODY BACKGROUND= /IR/IR-Images/bg.gif>
<IMG ALIGN="MIDDLE" SRC="/IR/IR-Images/ir-home.gif" ALT="ROME LABORATORY
BANNER"><BR>
<p>
<center><font size=8>
<!--Name-->
</font></center>
<br><br>
<!--Description goes here-->
<p><br><br>
<b><h4>Contracts:  </h4></b>
<ul>
<!-- LIST CONTRACTS HERE - Eventually we'll link to data sheet for each effort -->
<li>
<li>
</ul>
<b><h4>In-house:  </h4></b>
<ul>
<!-- LIST IN-HOUSE EFFORTS and Other Responsibilities (e.g., POC for whatever) -->
<li>
<li>
</ul>
<b><h4>Publications:  </h4></b>
<ul>
<!-- LIST PUBLICATIONS HERE -->
<li>
<li>
</ul>
<b><h4>Point Papers:</h4></b>
<ul>
<!--LIST POINT PAPERS HERE-->
<li>
<li>
</ul>
<p>
<img src="/RL-Images/RL-BAR.gif">
<P><h5><FONT SIZE=5>Rome Laboratory </FONT><img ALIGN=top src="/RL-Images/rl-ball-
blue.gif"> 32 Hangar Road <img ALIGN=top src="/RL-Images/rl-ball-blue.gif"> Rome, NY <img
ALIGN=top src="/RL-Images/rl-ball-blue.gif"> 13441-4114</h5>
<pre>
<p><h4><img align=bottom src="/RL-Images/mail.gif"><A HREF="mailto:
<!---E-mail address--->
@rl.af.mil">
<!---Name--->
</A><nobr>      <img align=bottom src="/IR/IR-Images/left_point.gif"><a
href="/IR/IRR/IRRE/IRRE_roster.html"> People page</a><nobr>      <img align=bottom src="/IR/IR-
Images/left_point.gif"><a href="/IR/IRR/IRRE/IRRE_home.html">IRRE Home Page</a>
</h4></pre>
<p>
<i> last updated:
<!---Today's date--->
</i>
<P>
<IMG SRC="/RL-Images/RL-BAR.gif"><BR><P>
<A HREF="http://www.rl.af.mil:8001/"><IMG SRC="/RL-Images/rl-button.gif" ALT="RL"></A>
<A HREF="/index.html"><IMG SRC="/RL-Images/ir-button.gif" ALT="IR"></A>
```

</BODY></HTML>
The speed given by the use of templates cut a great deal of time. If one had to create all of these HTML documents, it would be a huge, time-consuming, complex job with mark-up coding.

The homepages themselves, as is visible in the above template, were created using HTML 2.0, which is accessible by nearly all graphics-supporting browsers in use today. This was intentional, for the branch wished the pages to be easily accessible by as many people as possible. This, however, does not mean that the pages are dull or boring. The pages had a few extra features added to them by using tags from what has been dubbed "Enhanced Netscape." This term simply refers to the extra tags used to put features into standard HTML documents. Using these tags, people can install counters to keep track of how many people visit their page, link their e-mail address directly to a mail tool so that people can contact them directly from their homepages. Other tags used included the one that allows a person to place a background on their page and a tag that allows the coder to change the size of the text without using a header tag. The background is a simple graphic that is repeated to give each homepage a "texture" underneath the page's text and images.

ENHANCED NETSCAPE TAGS

Tag	Effect
<BODY BACKGROUND="picture.gif">	Produces a background behind the text.
<A HREF="mailto:johndoe@anywhere.com>	Allows a user to send e-mail directly from the WWW page.
<fontsize=number between 1 and 8>	Allows the coder to set the size of the text in a document. One is the smallest size, 8, the biggest.
<img src="/cgi-bin/Count.cgi?df=addams.dat|dd=D|ft=3">	Allows a person to keep track of how many people visit their page within a certain time.

The creation of the pages came next, where the aforementioned template was used. First, the basic HTML homepage document was created, edited, and set aside. The links were not established on the pages, for the need for a new directory on the server was apparent, and the name for the new path had not yet been determined, so the point paper "HTMLization" was completed. After the new directory had been created,

the links were established and checked for accuracy simply by looking at the syntax, for the link to the different URLs, or Universal Resource Locators will not function unless placed on the server.

Graphics showing the person superimposed over the Rome Laboratory logo were just one of the things added to the RL/IRRE website. Using Adobe Photoshop 3.0, this seemingly difficult task became relatively easy as a result of the many versatile editing tools available in the program. Creation of the images, now in JPEG format to preserve the color palettes, was essentially a four step procedure. First the photograph was taken using a digital camera and placed on a computer using a direct link from the camera's digital adaptations to the receiving computer via a SCSII interface. The images were then electronically mailed to the Image Exploitation Branch where the photographs were edited by removing the solid background and saving the pictures as separate Photoshop 3.0 files. These files were then pasted as an extra layer on top of a prefabricated Rome Laboratory background and saved as PICT files with a single layer. Using GraphicConverter, the PICT files were then changed to JPEGs. A GIF format was experimented with, but to maintain quality in the photographs, a JPEG format was necessary, for the 256 indexed colors of the GIF format reduced the clarity and the overall quality of the pictures. (See attachment 1.)

A similar use of templates was also attempted to be used with the point papers. The papers were in a Microsoft Word format, which is not compatible with WWW browsers (Netscape, Mosaic, etc.). The original idea was to make a template to simply fill in with the paper's information. However, this proved to be impossible, for many of the papers were of a different layout. This is because by using papers that may be older than others, the layout in which they were created in had been changed. This is why it was found to be easier to simply insert the tags manually, and then save the whole document with an HTML extension in a Microsoft Word Text Only format. This made the file fully compatible with the RL/IRRE server. Other papers were immediately converted into a SimpleText format, where the extra time spent saving as a Text Only document was eliminated. This enabled one to essentially work a little bit faster, but the extra editing options found in Microsoft Word proved to be very helpful during the HTML coding process.

The conversion of the point papers was two step procedure. First, the layout and overall look of the page had to be determined, and second, the actual placement of HTML tags had to be accomplished. The first step was considered, and it was decided that the point papers should have the same general look of all the other pages in the IRRE website. Therefore, several graphics were downloaded and placed at the beginning and end of all point papers so that the branch would have a uniform look throughout its website. The placement of the HTML tags was time consuming, but not complex, for the point papers follow a format similar to that of an outline used for the writing of a longer paper. The majority of tags placed were those of unordered lists (<ul> and) and those of the actual list items (<li>). The other tags included were the tags that supported the images and the font characteristics. There are also the occasional use of the <head> and <body> tags that make it easier to keep track of what part of the HTML document you are working on. These point papers were then saved as text and renamed (to keep the originals intact) and linked to the respective person's homepage. The following is a sample of the "typical" point paper HTML document written by a member of the Image Exploitation Branch, which has now been "HTMLized" through the placement of tags:

A Sample RL/IRRE Point Paper with HTML Tags

```
<html>
<title> ON PARALLEL AND DISTRIBUTED ALGORITHMS FOR HIGH-SPEED IMAGE
PROCESSING POINT PAPER</title>
<center><img src="ir-home.gif"></center>
<br>
<center><h3>POINT PAPER  ON PARALLEL AND DISTRIBUTED ALGORITHMS
FOR HIGH-SPEED IMAGE PROCESSING</h3></center>
<br><img src="IRbar.gif"><br>

<ul><li>-       The purpose is to develop parallel and distributed image
        processing algorithms to take advantage of parallel architectures
        to achieve shorter processing times.

<li>-   The effort is contracted through the Expert Science and Engineering
        Program to the University of Notre Dame.

<li>-   The contract is for 18 months and is expected to start Aug 96.

<li>-   The parallel architecture of focus is a heterogeneous cluster of
        networked UNIX workstations.

<li>-   Specific areas of algorithm focus are:

        <ul><li>--      Parallelize the rendering of an image on the monitor;

        <li>--  Taking advantage of hierarchical memory;

        <li>--  Performing load balancing techniques;
```

-- Investigating a shared memory architecture.

- The algorithms will utilize the Message Passing Interface (MPI) standard for communication.

- The algorithms are to be incorporated into a portable and extensible Parallel Image Processing Toolkit.
- This architecture will take advantage of free CPU cycles on existing workstations.

<img src="IRbar.gif">

Todd Howlett/RL/IRRE/DSN 587-4592/2 July 1996</html>

Attachment 2 is an example of a sample point paper as viewed on IRRE's website.

After all of the homepages, point papers, and images had been finished, and cleaned up, they were sent to the RL/IRRE server. There, the homepages and the graphics were separated from the point papers and placed into their respective directories. The next few days were spent going through the website and checking for any mistake, misspellings, and broken links, so that they would be functioning properly by the time my tour was up. At this point, all links are established, and all of the graphics are working. If you wish to view this site, it can be found at the following locations:

http://www-ir.rl.af.mil/IR/IRR/IRRE/IRRE_roster.html

http://www-ir.rl.af.mil/IR/IRR/IRR_home.html

In conclusion, the Image Products Division, Image Exploitation Branch now has a newly updated roster, complete with images of the employees, their publications, and their current point papers concerning the ongoing projects being worked on by RL/IRRE. With the release of these pages onto the World-Wide Web, the public can now follow the efforts of the Image Exploitation Branch as they make progress in the future. This can help to build support through grants and joint efforts that will propel the Image Exploitation Branch into the next century at full speed.

Recommendations:

As a result of briefing and demonstrating the new IRRE website, the Intelligence and Reconnaisance Directorate Technical Director, Mr. Joseph Camera, has with my help, initiated the development of a database for the management ond storage of current contracts, point papers, and publications to be used by the entire directorate in the future.

Attachments:

1-Sample IRRE Personnel Page as viewed with Netscape 1.1

2-Sample IRRE Point Paper as viewed with Netscape 1.1

RESOURCES:

Brandon, Bill, et al.Special Edition Using HTML, Second Edition. IN: Indianapolis, Que Corporation, 1996.

ATTACHMENT 1

Intelligence and Reconnaissance Directorate

Richard J. Simard

Richard is a Electronics Engineer with the Image Exploitation Branch, Image Systems Division (IRRE), Intelligence and Reconnaissance Directorate, of Rome Laboratory, Rome, NY. He received a Associate in Applied Science in Electrical Technology from Mohawk Valley Community College, a Bachelor of Science in Electrical Engineering from University at Buffalo, a Master of Science in Management Science from University Center at Binghamton and is currently pursuing Ph.D. studies in Systems Science from University Center at Binghamton. Richard's research interests include Image Intelligence (IMINT) System Design & Processes, Modeling & Simulation, Uncertainty and Chaos Theory and Applications.

Contracts:

- Secure Image Ciphering Based on Chaos
- Parallel Real-Time Video Encryption
- Multi-Frame Integration For Extraction of High Resolution Still Images from Video Sequences
- Simulation Based Optimization Environment
- Statistical Modeling with Imprecise Probibilities
- Imagery Intelligence (IMINT) Mix Study
- Joint Intelligence Suveillance and Reconnaissance (ISR) Architechture Evaluation Process
- Joint ISR Systems Engineering Technical Assistance
- Tactical Airborne Reconnaissance Architecture Trade-Off Study
- Battle Damage Assessment Modeling & Analysis
- Battle Damage Assessment (BDA) Intelligence System Modeling

In-house:

- Rome Laboratory Focal Point for Modeling & Simulation
- Air Force Office of Scientific Research (AFOSR) Summer Research Program Mentor for Image Exploitation Branch
- Summer Employment Research Program Mentor for Image Exploitation Branch
- Member of Air Force Studies And Analysis Technology Working Group on Modeling and Simulation

Publications:

- Simard R. J., "A Simulation Development Process - Simulation EnVironment" Proceedings of the 6th Annual Conference on AI, Simulation and Planning in High Autonomy Systems, Mar 96
- Simard R.J., Mohr G.C., Richey K., Hite D., "Science And Technology Comprehensive Modeling And Simulation Plan (STCMSP)" Published by Air Force Material Command Jul. 93
- Simard R.J., "Systems Design, Modeling and Metamodeling" International Journal of Systems Practice, Aug. 93
- Simard R.J., Zielger B.P., Couretas J.M., "Verb Phrase Model Specification via System Entity Structures" Proceedings of the 5th Annual Conference on AI, Simulation and Planning in High Autonomy Systems, Jun. 93

- Stevenson R.L., Schultz R.R., "Multiframe Integration For High-Resolution Video Stills" RL-TR-96-21, Feb. 96 (Simard R.J., Technical editor and oversight)
- Roark W., Bass J., Gossage B., Christian J., White K., "Battle Damage Assessment Simulation (BDASIM) and Study" RL-TR-96-19, Feb. 96 (Simard R.J., Technical editor and oversight)

Point Papers:

- Point Paper on Secure Image Cyphering Based on Chaos
- Point Paper on High-Performance Simulation-Based Optimization Environment
- Point Paper on Multi-Frame Integration for the Extraction of High Resolution Still Images from Video Sequences
- Point Paper on Statistical Modeling with Imprecise Probabilities
- Statistical Modeling with Imprecise Probabilities
- Point Paper on Battle Damage Assessment (BDA)
- Point Paper on Rome Laboratory Support to C4 Intelligence Surveillance Reconnaissance (ISR) Architechture Modeling
- Point Paper on Joint Intelligence Surveillance and Reconnaissance (JISR) Architecture Evaluated Process

---

Rome Laboratory • 32 Hangar Road • Rome, NY • 13441-4114

Richard J. Simard People page IRRE Home Page

last updated: 29 JUL 96

IR

RL IR

ATTACHMENT 2

Intelligence and Reconnaissance Directorate

ROME LABORATORY WORK SERVER

POINT PAPER ON SECURE IMAGE CIPHERING BASED ON CHAOS

IR

- This is a GRANT 6.2 research effort with the State University of New York at Binghamton, Center for Intelligent Systems. Dr. Jiri Fridrich is the Principal Investigator.
- **Technology:**
  - A discretized form of a chaotic, invertible, algorithm is iteratively applied to a lattice of pixels to permute the pixels and their gray levels in a complex manner.
  - Localized information in the original image is distributed in the encrypted image in an apparently random fashion, but there is order in chaos.
    - -Invertibility of the maps makes the deciphering possible.
    - -The parameters of the maps serve as a ciphering key.
    - -The simplicity of the chaotic maps makes the ciphering/deciphering process fast.
    - -There is no information loss in the ciphering/deciphered image.
- **Security:**
  - The security of the cipher is guaranteed by the sensitivity to initial conditions, chaotic parameters, and by the large number of possible keys.
  - The histogram of the enciphered image is modified by combining the permutation of pixels with gray levels.
  - The number of all possible ciphering keys depends on the number of pixels in the digital image.
  - An image consisting of 512 x 512 pixels, will have more than 10128 ciphering keys.
  - -Time to search through all possible ciphering keys for a 256 x 256 image is 1036 seconds, or more than 1028 years.
- **Applications:**
  - Chaotic ciphering has the potential to provide a low cost method for transmission of imagery, video, text for both government and commercial applications.
  - Application for a patent for this work is in progress.

IR

Mr. Richard Simard/RL/IRRE/DSN 587-4591/24 Jul 96

INFORMATION ON THE INTERNET and PEM TEST CIRCUIT DESIGN

Daniel Grabski

Holland Patent High School
Thompson Road
Holland Patent, NY 13354

Final Report for
High School Apprentice Program
Rome Laboratory
Rome, NY

Sponsored by
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory
Rome, NY

August 1996

INFORMATION ON THE INTERNET and PEM TEST CIRCUIT DESIGN

Daniel Grabski
Holland Patent High School

Abstract

I was involved in both software and hardware projects for my research. I wrote HTML code for the Rome Laboratory Electromagnetics and Reliability World Wide Web (WWW) pages. In writing these pages, many features of Netscape-type browsers were used, including the ability to accept JavaScript commands and the ability to create frames. By editing MoleculeViewer, a Java program for displaying and rotating a molecular model, I was able to create a realistic model of a molecule that can be rotated and viewed on a WWW page. For hardware research, I utilized a LTX77 Automatic Test System to test LM2904 dual opamps from different manufacturers for in-house plastic encapsulation microcircuit (PEM) research. I also designed a circuit and program to be used with the LTX77 to test up to four LM324 quad opamps at a time to collect data for field PEM research in collaboration with the Army.

INFORMATION ON THE INTERNET and PEM TEST CIRCUIT DESIGN

Daniel Grabski

Introduction

In order to make the most of my summer research, I worked on some projects which involved software, and some which involved hardware. For that reason, my report consists of two sections. The first section deals with my software projects; the second section deals with my hardware projects.

Software -- Information On The Internet

What we call the Internet today began as ARPANET. This was a network designed primarily to connect universities, the military, and defense contractors so they could share information. Since ARPANET was designed and developed in 1969, the Internet has evolved from other networks being created, growing, splitting, and combining. The Internet today consists of many interconnected networks that can communicate with one another.

Today, millions of people in the world are connected in some way to the Internet. Any person with a computer, a modem, and an Internet service provider can access the Internet. The number of households with computers is rising constantly. Modems are very affordable -- one can buy a 28,800 bits per second modem for as low as $150, and that price is even dropping. With numerous commercial service providers, like Compuserve and Prodigy, offering its customers Internet access for $10 a month with a local telephone call, it is no surprise that more people are using the Internet every day.

With this means of communication to millions of other people in the world, a system of distributing information to these people has emerged. Through Hypertext Transfer Protocol, or HTTP, any Internet user can create interactive documents using Hypertext Markup Language, or HTML. These pages can then be accessed by other Internet users through a WWW browser. Many companies offer free browsers, like Lynx, a popular UNIX text browser, and Netscape, a graphical browser. Most newer graphical browsers can display images, background images, different fonts, and different sizes and colors of text. This means that companies and individuals have a user-friendly and interactive method of

distributing information around the world. I incorporated some new features of these browsers into my software project to help distribute information.

I created new WWW pages for the Electromagnetics and Reliability Directorate at Rome Laboratory. My main goal was to utilize features of new graphical browsers to enhance the appearance of the pages, and to help the user navigate through the pages in order to distribute information about the Directorate. To reach these goals, I used the frame capability of most browsers and the browser's ability to accept JavaScript code.

The main Electromagnetics and Reliability page consists of three frames created by the FRAMESET tag in the HTML code. The frame definition is basically in the following format:

```
<FRAMESET rows="30%,70%">
      <FRAME name="frame1" src="frame1.html">
      <FRAME name="frame2" src="frame2.html">
</FRAMESET>
```

The frame definition starts with the `<FRAMESET rows="30%,70%">` tag. This identifies the width of the rows created by the frame. This would create two frames; one that is 30% of the height of the browser window, and one that is 70% of the height of the window. The percentages can be replaced by actual sizes in pixels. The word "rows" can also be replaced by "columns", if vertical columnar frames are desired. Then, the `<FRAME>` tag identifies information about the individual frames. The `name` section of the tag identifies the frame's name, and the `src` section identifies the HTML source for the frame. Then, the `</FRAMESET>` tag identifies the end of the frame definition. This allows different information to be displayed in each frame.

As mentioned before, the home page consists of three frames. The frame on the top is the largest; it displays the main information. The second frame is the smallest; it displays an informational message about where the user is among the Directorate's web pages. The bottom frames contain a number of buttons that take the user to different web pages in the Directorate. The bottom frame is the most important, since it gives the user one-click access to any major part of the Directorate's web pages without having to scroll through the main page. Because of the use of frames, the informational message and the navigational buttons are always available to the user. This makes navigation through the pages

easier, because the user will never have to scroll through a document to get back to the main page or a main section of the pages.

However, a problem arises when trying to display new information in two or more frames (or even a different frame) as the result of one action. Normally, if a link is clicked in one frame, the new page will be displayed in that same frame. There was no method of displaying information in other frames, unless JavaScript commands are used.

JavaScript is an open cross-platform object scripting language created by Sun Microsystems. JavaScript was designed to create live applications for the Internet. It can be integrated into a Web page, where JavaScript applications can be written to interact with the user, perform a function, and display information to the user. JavaScript code is entered as part of an HTML document, and is interpreted by the Web browser.

I used JavaScript in the Web pages to overcome the problem of calling information to more than one frame at a time. JavaScript has the capability of calling two or more different HTML documents to different frames with one action. For example, the following code would be included with the original HTML document:

```
<script language="JavaScript">
<!--
function ReturnToMain()
      {
      this.open("~/options.html", "options_frame")
      this.open("~/welcome.html", "main_frame")
      }
// -->
</script>
```

This would open the document "options.html" in the frame named "options_frame", and the document "welcome.html" in the frame named "main_frame". Then, to create a button that would call this function and do these actions, the following would be inserted into the body of the HTML document:

```
<form name="buttons">
      <input type="button" name="option_button"
      value="Click this button."
      onClick="parent.ReturnToMain()">
</form>
```

This would create a clickable button with the text "Click this button." on it. When this button was clicked, the browser would call the function ReturnToMain() to the object's parent, the browser window. The document "options.html" would appear in the "options_frame" frame, and the document "welcome.html" would appear in the "main_frame" frame.

This basic method was used throughout the pages to open more than one document in different windows. JavaScript was also used to call the different documents used in the informational frame, which informs the user where he or she is. JavaScript was used in the organizational chart in order to display other web pages and to graphically show the organizational structure within the Electromagnetics and Reliability Directorate.

The use of frames and JavaScript, along with basic HTML elements, like form buttons and links, were combined to assist users navigating these pages, and to enhance the pages' appearance. The final product of this work can be found at *http://www-er.rl.af.mil/~grabski/*.

I was also involved in adapting a Java application to display a molecular model from a molecular dynamics simulation. Java is a programming language created by Sun Microsystems and developed to be completely platform-independent, provided that the computer contains the Java "virtual machine" -- a platform designed to run on the host computer's operating system and interpret the bytecodes from a Java application. The Java "virtual machine" has been ported to many different platforms, and has been integrated with browsers like Netscape to allow the browser to run Java applications. Because of its platform independence, Java is ideally suited to the Internet, which is inhabited by many different platforms, from IBM mainframes and Sun servers down to home PCs. With the Java language, a company can write and compile an application once in Java. It can then distribute it, or place it onto a WWW page. Then, any computer that has the Java virtual machine can run the application, without having to port and recompile the application for the new platform.

One application using the Java programming language is MoleculeViewer. It comes with the Java Developer's Kit from Sun Microsystems. It takes a file with data about the positions of atoms in a molecule. It then creates a visual representation of the model that the user can rotate and view. This application can run on its own, or be integrated into a WWW page, where a Web browser would

interpret the application's bytecodes and run the application. In our case, the output would be placed into a WWW page.

The Java language allows information to be distributed through HTML in a new way. Instead of having static links or pictures on a page, Java allows individuals to create applications that can distribute information, or be distributed on the Internet, with no fear of incompatibility with a particular platform.

The MoleculeViewer application took data from the data file about the positions and types of atoms in the molecule. It then put the position data into a matrix, scaled the data to fit in the screen, created images of the individual atoms, and then displayed the finished molecule. In order to rotate the molecule, the application used the mouseDown and mouseDrag functions to get the location of the mouse pointer and if the mouse was being dragged across the screen. The application then translated the position matrix according to how much the mouse was dragged and redisplayed the molecule. After adjusting the spacing of the atoms, the model looked realistic. The final product is available at *http://www-er.rl.af.mil/~grabski/atom/*.

Hardware -- PEM Test Circuit Design

As part of a field plastic encapsulated microcircuit (PEM) research program with the Army, Rome Laboratory will be receiving 750 LM324 and LM124 quad operational amplifiers for testing at various temperatures using a LTX77 Automatic Test System (ATS). In the past, dual opamps of this type were tested one at a time, using relays to switch between the two opamps in the package. The test circuit used six relays, and testing of the packages became time-consuming for a large number of packages. This would pose two significant problems for testing the quad opamps. First, the amount of test circuitry involved would become excessive, and second, the time involved in testing 750 opamps over temperature (-55 to 125 degrees Celsius) one at a time would be unreasonably long. A new method of testing the opamps in a reasonable amount of time and with a reasonable amount of circuitry had to be created.

In order to reduce the time involved in testing all 750 opamps, my goal was to create a circuit that would accept four packages at once and test each opamp in each package in succession. If the circuit that was used for the dual opamps previously was copied eight times to accommodate four quad opamps,

the circuit would use a total of 48 relays, and eight more sets of resistors, capacitors, and nulling amplifiers. This is far too much equipment that could potentially cause error in the measurements.

However, the capabilities of the LTX testing station can reduce the number of circuit components drastically. The testing circuit consists of a few basic elements: a grounded supply, the device under test, a nulling opamp, and the output. (See Figure 1.)

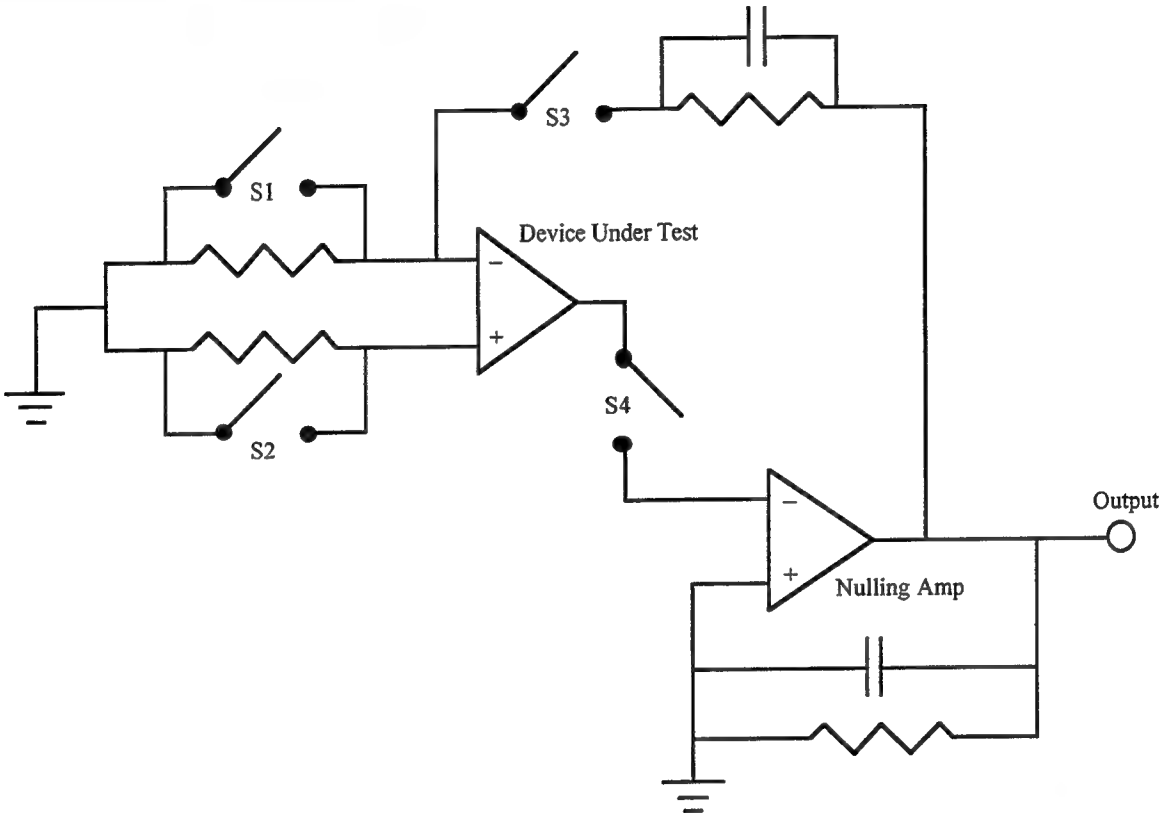


Figure 1. Original Test Circuit

The switching was done by the relays, which would switch between the two opamps under test in the package. For the new circuit, I used the LTX TS88 matrix to perform the switching between the opamps under test. The matrix consists of 8 "lines" and 96 "pins" that can be interconnected in any combination imaginable. For example, line 5 can be connected to pins 3, 8, and 24. In this way, the new circuit could have only one grounded supply and one nulling amp, which would be connected to lines, for all sixteen opamps in the four packages, which would be connected to pins. The LTX program would then take care of switching the connections between the lines and pins.

To eliminate the use of relays to bypass the input resistors, a grounded supply was connected in parallel to four lines, two of which had input resistors, and two that did not. The opamp inputs are then connected to pins. The lines and pins are then connected, but only one line is connected to any one pin at a time. So, to bypass both input resistors, one of the lines without a resistor would be connected to the inverting input of the opamp, and the other line without a resistor would be connected to the noninverting input. So, by using four lines for the different input combinations needed, and a pin for each opamp input, all of the relays on the input can be eliminated.

To eliminate the relays connecting the opamp under test to the nulling amp, two more lines were used. They were connected on the sides of switches 3 and 4 (see Figure 1) to replace those relays. Then, pins were connected in place of those relays on each opamp. So, the ATS can connect the two lines from the nulling amp to whatever opamp is under test. This eliminated the remaining relays.

An added bonus from the use of the LTX TS88 matrix for this circuit is that no external circuitry to the device under test is duplicated. Therefore, the chance of malfunction in the circuit is greatly reduced, and accuracy is improved.

Conclusion

I highly enjoyed the work that I did on both of these projects. I was able to participate in projects that had an effect on the real world. The work on the Electromagnetics and Reliability pages will assist Rome Laboratory in getting information out to universities and other companies that could benefit from technology transfer with Rome Laboratory. The PEM research has become increasingly important lately with the military. In the past, all integrated circuits used in the military would be hermetically sealed in their ceramic and metal packages to isolate the microchip from harmful environments, like heat and humidity. However, hermetically sealing every microcircuit used is no longer feasible. Integrated circuit manufacturers have reduced interest in producing hermetically sealed chips because of the cost involved, and because they primarily produce plastic encapsulated microcircuits for consumer applications. Therefore, the military is looking at PEMs as a much cheaper alternative to hermetically sealed microcircuits. My circuit design will be used to collect data from PEMs that have been undergoing stress-testing to determine the reliability of PEMs in military environments. This research could possibly

save the military thousands of dollars in integrated circuits. In conclusion, I feel that my work this summer was very worthwhile.

MEMORIES OF THE FUTURE:
A STUDY OF BIT-ORIENTED OPTICAL MEMORY

Nicholas Hrycan

Thomas R. Poctor Senior High School
Hilton Avenue
Utica, NY 13501

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

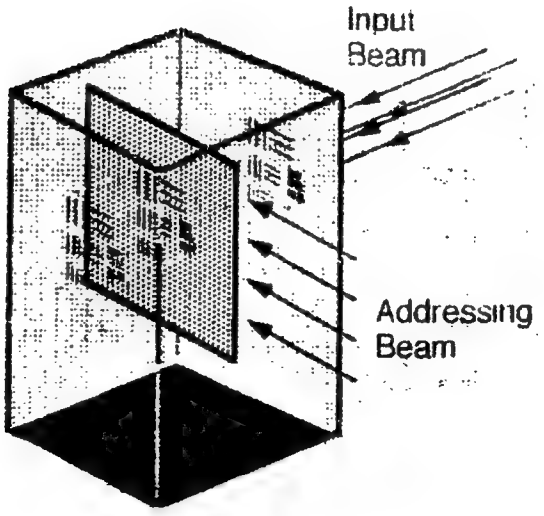
August 1996

MEMORIES OF THE FUTURE:
A STUDY OF BIT-ORIENTED OPTICAL MEMORY

Nicholas Hrycan
Thomas R. Proctor Senior High School

Abstract

Aspects to consider when developing a new memory system are capacity, access time, data transfer rate, storage persistence time, and cost per megabyte. Current two-dimensional (2D) memory devices store information as bits on a flat surface called a bit plane. As a result, this information is retrieved from the bit plane bit by bit. On the other hand, three-dimensional (3D) optical memory devices take it a step further by storing the information by entire bit planes and stacking them in the third dimension. Consequently, one memory operation is performed on the entire bit plane, resulting in a tremendous memory bandwidth increase over existing 2D memories. The figure below shows one of the bit planes in the cube. Image storage and retrieval is one area where improved storage technologies are required. This paper will present Bit-Oriented (a.k.a. two-photon) optical memory by breaking it into Theory, System Operation, and Problems and Possible Solutions.



MEMORIES OF THE FUTURE:
A STUDY OF BIT-ORIENTED OPTICAL MEMORY

Nicholas Hrycan

Introduction to Optical Memories

Here are some quick facts on Bit-Oriented 3D optical memory:

- Each bit occupies a specific location in 3D space
- Information stored using amplitude recording media
- Process utilized to store information is two-photon absorption
- Capable of processing data in parallel
- Potential capacity of 1 Terabit (an equivalent of 200 CDs)
- Potential data density on the order of $1Tb/cm^3$
- Potential throughput 1 Gigabit/sec
- Potential access time of nanoseconds
- Currently is a ROM (Read Only Memory) but the ultimate goal is Read/Write/Erase

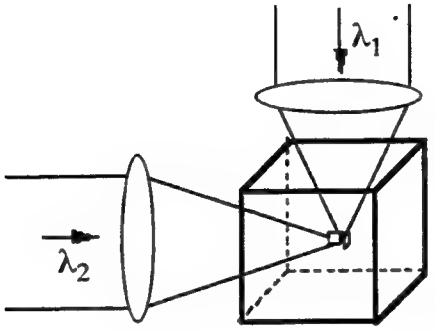
Bit-Oriented 3D Optical Memory

THEORY

Bit-Oriented optical memories utilize two-photon absorption technology to record information. Two-Photon absorption technology, as its name suggests, is the simultaneous absorption of two photons, whose combined energy is equal to the energy difference between the initial and final states of the recording material, which causes a change in its molecular structure. The induced change can be detected as a change in the absorption spectrum, fluorescence, index of refraction, or electrical properties of the material.[1] The two distinct photo-induced forms, the excited and original, act as the '0' and '1' for information storing and retrieving. For example, the presence or absence of fluorescence is detected and classified as a '0' or '1' respectively for the memory bit. Therefore, by intersecting two optical beams, the material's optical properties can be altered locally and addressed anywhere within a three-dimensional space. Data readout is accomplished by probing the material with a single beam to measure a change in one of the material's optical properties.

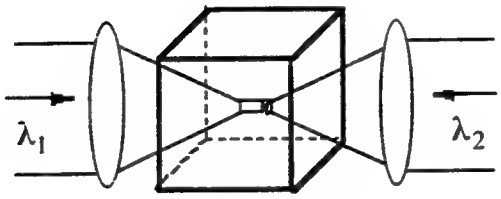
In two-photon absorption there are two different ways the optical beams can intersect in the material. These two methods are called Orthogonal Beam Addressing and Counter-Propagating Beam Addressing. In orthogonal beam

addressing, the two beams enter the cube at a right angle to each other and write at the spot where they meet. The volume of intersection is determined from the beam cross-section at the point of insertion. This volume can be reduced by using higher numerical aperture lenses or by reducing the optical laser wavelengths. In counter-propagating beam addressing, two short-pulsed laser beams are directed into opposite faces of the cube and write the information where they meet. The minimum bit volume is determined by the smallest of the diffraction limited spots, as well as the pulse width of the colliding beams.[2]



(a)

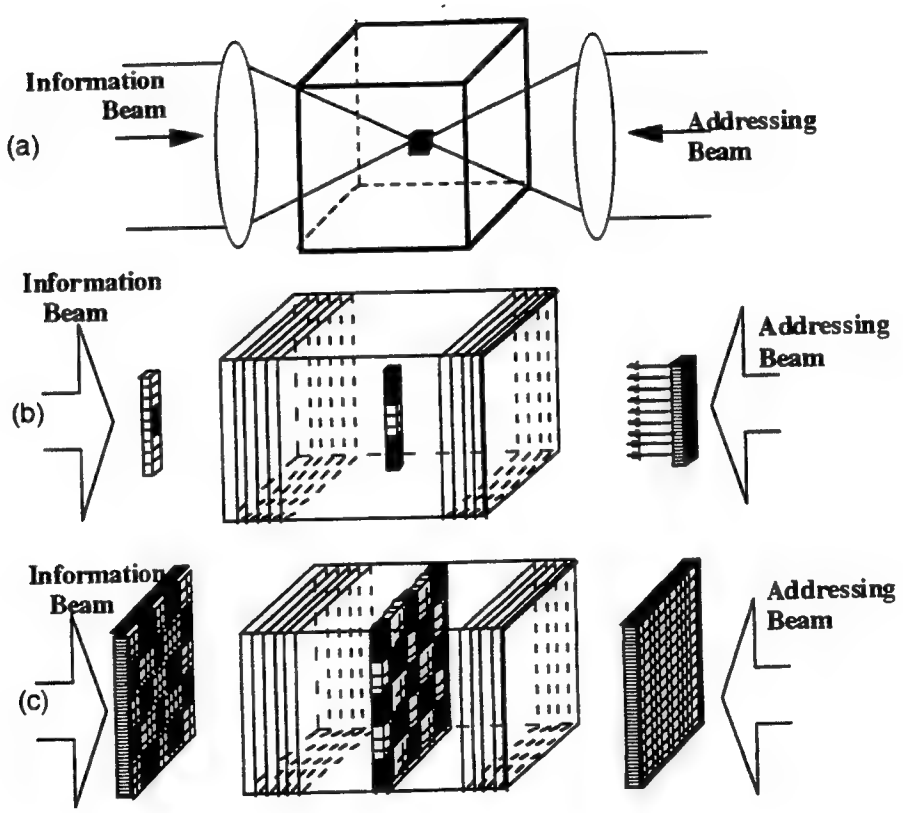
Orthogonal Beam Addressing



(b)

Counter-propagating Beam Addressing

In either of the beam addressing methods, the region of overlap of the two beams can be described via three different architectures: point (bit), line (row or column), and plane (cross-section of the material). The figure below shows the various data formats for two-photon memory using counter-propagating beam addressing. a) bit b) line c) plane.



Storage media ranging from lobster eyes to pond scum (bacteriorhodopsin) to NNA/Rhodamine B have been used in Bit-Oriented 3D optical memory. The characteristics being looked for in a two-photon recording medium are:

- Photochromism (The change of chemical structure after excitation by light, making possible the '0' and '1'.)
- A fluorescence in one of the two chemical states
- Ability to read data 10^6 times without loss of information
- High solubility of the material to create high concentrations needed for efficient two-photon process
- High quantum efficiency of the read form fluorescence
- A wide enough wavelength shift between the read beam and peak of the fluorescence spectrum to prevent cross-talk (bits of information interfering with each other)
- Sensitivity to low levels of light

It is difficult to find a material that possesses all of these characteristics. Below is a brief description of Spirobenzopyran, Bacteriorhodopsin, and NNA/Rhodamine B; which are a few materials that have been used in Bit-Oriented 3D optical memories. All of the materials below are estimated to cost less than a cent per gigabit.

Spirobenzopyran

Spirobenzopyran has two distinct forms which are called spiropyran and meracyanine. These two forms provide the two states necessary for storing information in the binary format. Erasure in spirobenzopyran can be achieved either by increasing the temperature of the material or by irradiating it with green light. By raising the temperature, the meracyanine (written) form is promoted above the energy barrier separating the two ground states, causing the meracyanine molecules to revert to the spiropyran form. Note that thermally the memory can be erased but not selectively erased. Irradiation via green light has been shown to erase specific regions or planes without affecting other areas of the memory. Unfortunately, spirobenzopyran has shown a limited lifetime of approximately 20 hours at room temperature for memory retention. This instability at room temperature has been found to be the result of an ion attraction between a O- and N+ in the written form which reverts the material to its original state. This destructive process has been thwarted by the addition of an acid (HCl) which prevents the oxygen and nitrogen ions from reacting. However, although the addition of the acid stabilizes the molecule, it eliminates spirobenzopyran's chances for secondary memory usage because the concentration of molecules per unit volume is smaller than necessary for high density information storage.[3]

Bacteriorhodopsin

This material is the light transducing protein found in the bacterium *Halobacterium halobium*. Two-photon absorption techniques can produce read-write-erase capabilities to create optical memory.[5,6] The bacteriorhodopsin has a light sensitive chromophore bound to the protein. Any change in the electronic environment of the chromophore binding site results in a change of the spectral characteristics of the overall protein. These changes can be induced by light absorption at proper wavelengths and by temperature changes resulting in a photocycle. Advantages of this material are the high photocyclicity exceeding 10^6, high quantum yield, and near room temperature operation. The bacteriorhodopsin memory can be bulk thermally erased and selectively erased by illumination with blue light. Unfortunately, the readout process is destructive, requiring the memory to be rewritten after each use. [2]

Nitronaphtialdehide/Rhodamine B

NNA/Rhodamine B consists of two materials: an acid generator and a dye precursor. The acid generator is a two-photon sensitive molecule which becomes an acid when excited by light. The dye precursor is another molecule which reacts with the acid to generate a room temperature stable, strongly fluorescing dye material. The write form is a mixture of the acid generator and dye precursor and the read/written form is the fluorescing dye material that is the end result of the chemical reactions that take place after photoactivation. Benefits yielded include the observations that a) the read and write forms have shown long term (years) stability at room temperature and b) the process is more efficient demonstrating higher signal-to-noise ratio at lower laser power. The problem with NNA/Rhodamine B, as with the above mentioned materials, is that it is still not sensitive enough to light and consequently demands an impractical amount of power from the laser.[3]

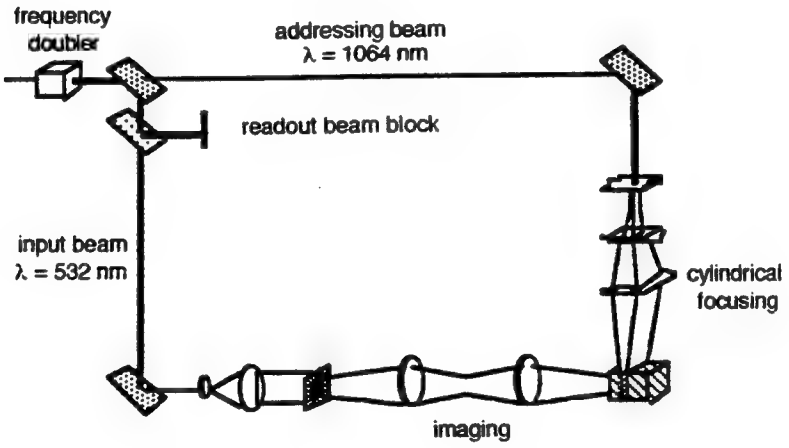
SYSTEM OPERATION

The unit at Rome Laboratory uses a HeNe laser to read the 9mm X 9mm X 13mm cube, although the cube was written with Nd:YAG laser. A stepper-motor driven stage moves the volumetric ROM so that each image plane is illuminated by the HeNe laser. As the image planes are illuminated by the small HeNe laser, the CCD camera collects the fluorescence and displays the recorded images on a video monitor. There were 16 images (two-dimensional information) recorded, some of which were composed of pictures and others as patterns of light and darks spots which are translated as $1'^{S}$ and $0'^{S}$ to the computer.

The system below uses picosecond pulses of the first and second harmonic (λ_1 = 1064nm, λ_2 = 532nm) of a Nd:YAG laser. Less power is required to read the memory but much power is consumed to write because the chemical properties of the materials are being changed.

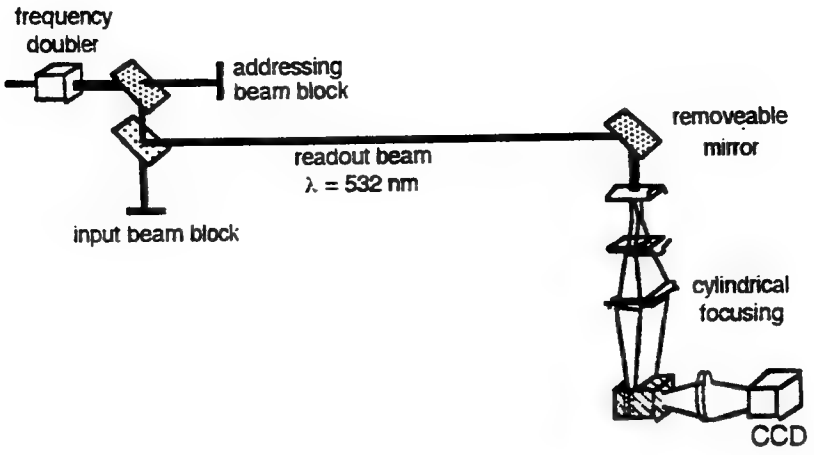
Writing[4]

Digital information is recorded in the two-photon material as pages of digital data, the data planes separated in the axial direction. The input/data arm of the system (λ_2 = 532nm) is sent to the Spatial Light Modulator (translates the laser into a pattern) and imaged to the proper plane within the cube. A second, addressing beam (λ_1 = 1064nm), propagating orthogonal to the input beam, is cylindrically focused throughout the cube. A page of data is recorded when the input beam and addressing beam simultaneously illuminate the image plane. The axial resolution is limited by the divergence of the addressing beam, while the lateral resolution is determined by diffraction effects and/or aberrations in the input arm.



Reading[4]

Readout is achieved by measuring the fluorescence of the material. The input beam is blocked and the 532 nm beam is cylindrically focused to read the proper data page. For the unwritten material, the 532nm beam is unabsorbed and passes through the system. However, for the written form of the material, the readout beam is absorbed, exciting a fluorescence at a longer wavelength. The readout plane is then imaged through a wavelength selective filter onto a CCD camera (detects image and translates it for the computer). Thresholding of the individual pixels of the CCD is performed to digitize the data.



PROBLEMS AND POSSIBLE SOLUTIONS

Laser

What makes Bit-Oriented 3D optical memory currently impractical is that the materials require lasers, such as Nd:YAG, to write efficiently. A Nd:YAG laser, for example, is undesirable because it:

- Is very expensive
- Has to be water-cooled
- Consumes too much power
- Is very big and bulky

A major goal of Bit-Oriented 3D optical memory is to use as little power as possible, just enough to change the chemistry of the material to record. These problems can be solved if the materials can be evolved into something that can incorporate existing laser diodes that are in CD players.

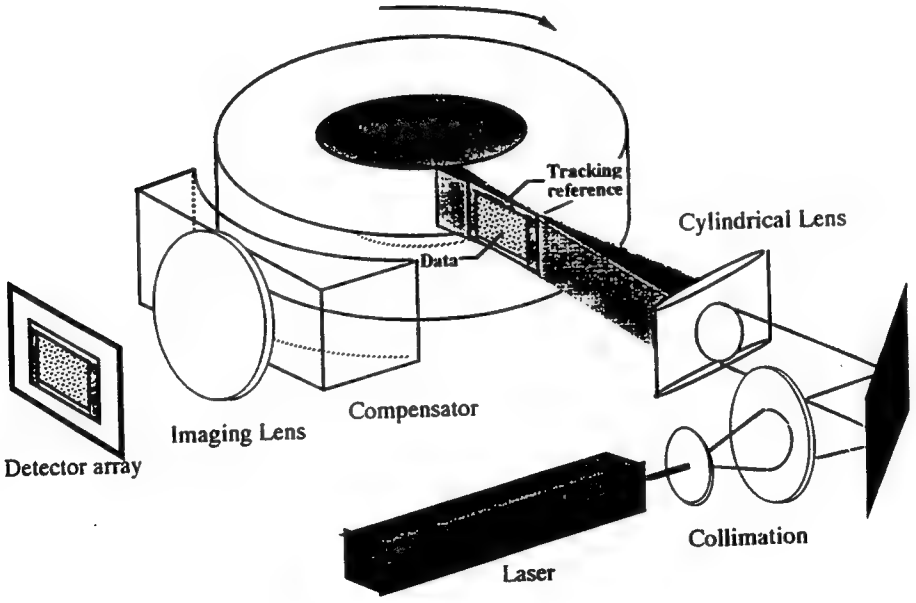
Cube Qualities

Besides sensitivity to low levels of light, important characteristics of the cube are surface quality and material homogeneity. Before the cube of the material can be used, it must be polished. This is due to the fact that an imperfect surface deteriorates the quality of the data as a direct result of the rough surface scattering the light. For example, if you had two cubes, one being better polished than the other, the better polished cube would have fewer errors made in the recorded data. In addition, the better polished cube has the added benefit of requiring less

power from the laser to record data. Generally, the cube is polished to a flatness of a 10th of the wavelength of the shorter beam. This is currently done mechanically but a change to injection molding of the material could eliminate the polishing process altogether. However, going to injection molding could create a problem with getting a uniform density of the material. Another problem with the cube is light bouncing within it, causing crosstalk. A possible solution to this would be an anti-reflection coating. This would reduce the glare and back-scatter in addition to increasing the efficiency of the cube by getting more of the light supplied into the cube.

Focusing

Since the cube is moved with a stepper motor to reach the various bit planes, there is a problem with the focusing of the material being that it is too slow and imprecise. An alternative approach to this revolutionary method of memory storage has been devised called the 3D-CD. The 3D-CD is a disk shape of the of the material that would be spun and the laser kept stationary. Refocusing would be eliminated because the medium would rotate planes of data in the stationary focal point of the laser. Just as is the case with the cube, the 3D-CD utilizes NNA/Rhodamine B as its recording media. However, the 3D-CD can not store a terabit of information as projected for the cube. This is due to the fact that there is wasted space between the information planes because it is recorded like spokes on a wheel.



Miscellaneous

Shock and vibration does not affect 3D memories when they are inactive, but it does affect them in the read and write mode just as the CD.

The three different architectures of beam overlap (bit, line, and plane) each have their own advantages and disadvantages. The plane format can collect a lot of information in a short amount of time, but unfortunately it requires more power from the laser and some storage capacity is lost. This is the result of the addressing beam diverging as it propagates, consequently requiring larger spacing between data planes. The point format doesn't have these problems, but the tradeoff is that it would take an unreasonable amount of time to do the job since it is done bit by bit. The line format falls somewhere in between; it is not as fast as the plane format but not as much storage capacity is lost.

Conclusion

All in all, when the obstacles of adequate materials, crosstalk, and focusing are overcome, three-dimensional bit-oriented optical memory will revolutionize memory storage systems with its data density of $1Tb/cm^3$, its throughput 1 Gigabit/sec, and its parallel data processing. Such a system would allow you to hold a 100,000 book library, the size of a sugar cube, in the palm of your hand.

References

[1] Dvornikov, A.S., and Rentzedis, P.M., "Two-Photon 3D Optical Memory," *IEEE Proc. of 1994 Conference On Solid State Memory Tech.*, p. 109-118, 1994.

[2] Jamberdino, A., Haritatos, F., Canfield, B., "Optical Memories for Large Data Base Storage," *Rome Laboratory Technical Journal, p.* 86-91, Dec. 1995.

[3] Esener, S.,Rentzepis, P., "Two Photon 3-D Memories", *Final Technical Report RL-TR-95-76,* Apr. 1995.

[4] Kann, J. Canfield, B., Jamberdino, A., Clarke, B., Daniszewski, E., Sunada, G., "Mass Storage and Retrieval at Rome Laboratory", *To be published in proceedings from the Ffth NASA/Goddard Confrence on Mass Storage Systems and Technologies,* Sept. 17-19, 1996.

[5] Birge, R.R., "Protein-based Optical Computing and Memories," *Computer*, p. 56-67, Nov. 1992.

[6] Brauche, C., Hampp, N., Oesterhelt, D., "Optical Applications of Bacteriorhodopsin and its Mutant Variants," *Advanced Materials 3*, p. 420-428, 1991.

Jamberdino, A., Haritatos, F., Canfield, B., "Optical Memories for Large Data Base Storage," *Rome Laboratory Technical Journal, p.* 86-91, Dec. 1995.

Esener, S.,Rentzepis, P., "Two Photon 3-D Memories", *Final Technical Report RL-TR-95-76* Apr. 1995.

Interview with Joshua L. Kann and Gary Sunada, Aug. 1996.

MAGNITUDE MEASUREMENT OF ELECTROMAGNETIC FIELD INTENSITIES USING AN INFRARED MEASUREMENT TECHNIQUE

Sandra L. Jablonka

Oneida High School
560 Seneca St.
Oneida, New York 13421

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Rome Laboratory

August 1996

MAGNITUDE MEASUREMENT OF ELECTROMAGNETIC FIELD INTENSITIES USING AN INFRARED MEASUREMENT TECHNIQUE

Sandra L. Jablonka
Oneida High School

Abstract

Electromagnetic (EM) field intensities were studied, using an IR (Infrared) measurement technique. Sensitive to EM radiation, Carbon loaded Kapton paper, was used as a detector. It increases in temperature with a direct relation to the intensity of the radiation. The radiation is absorbed by the detector which causes the ambient temperature at the surface of the paper to rise. An IR camera used in conjunction with AGEMA Thermovision software allows the various temperatures on the paper to be identified and to be displayed on a computer screen. The camera was placed on the opposite side of the paper as the horn. The paper was radiated with EM radiation from the horn antenna and the temperatures were measured using the IR camera. The purpose of the study was to quantify the errors that can affect the intensity and repeatability of the EM field.

MAGNITUDE MEASUREMENT OF ELECTROMAGNETIC FIELD INTENSITIES USING AN INFRARED MEASUREMENT TECHNIQUE

Sandra L. Jablonka

Introduction

The EM field intensities are being studied in hope of perfecting this new diagnostic technique. The new technique will allow phased array antennas, that are not working correctly, to be diagnosed on site. This will eliminate the cost and time of sending the antenna away to be fixed. The IR measurement technique is a thorough and efficient method of collecting data which can detect problems with the antenna. The number of potential sources of error which limit accuracy and repeatability have been investigated and will be discussed.

Setup

An IR camera setup was used to measure EM field intensities. It included a standard gain horn antenna, a lossy detector made of carbon loaded Kapton paper, and an IR camera. IR detectors of different surface resistance were used, one of 100ohms/sq and the other of 1500 ohms/sq. The lower the surface resistance, the more radiation was absorbed which caused the ambient temperature to rise higher. A Hewlett Packard Synthesized Signal Generator was used to provide a frequency of 4GHz. This signal was sent to a pre-amplifier. This can amplify frequencies between 2.0 - 4.0GHz and has a power output of

2 Watts. Then it was amplified by a Logi Metrics TWT 200 Watt amplifier. The signal was sampled by a Bi-directional Coaxial Coupler. Power meters were connected to the coupler to monitor the forward and reflected power at the antenna. The horn antenna created the EM field that was incident on the detector, which was set in a wooden stand. It had one stationary disc and one rotating disc. This made it easy to rotate the screen to an angle. The IR camera, which was positioned on the opposite side of the screen, viewed the temperatures as the radiation penetrated the detector and transmitted the information to the computer.

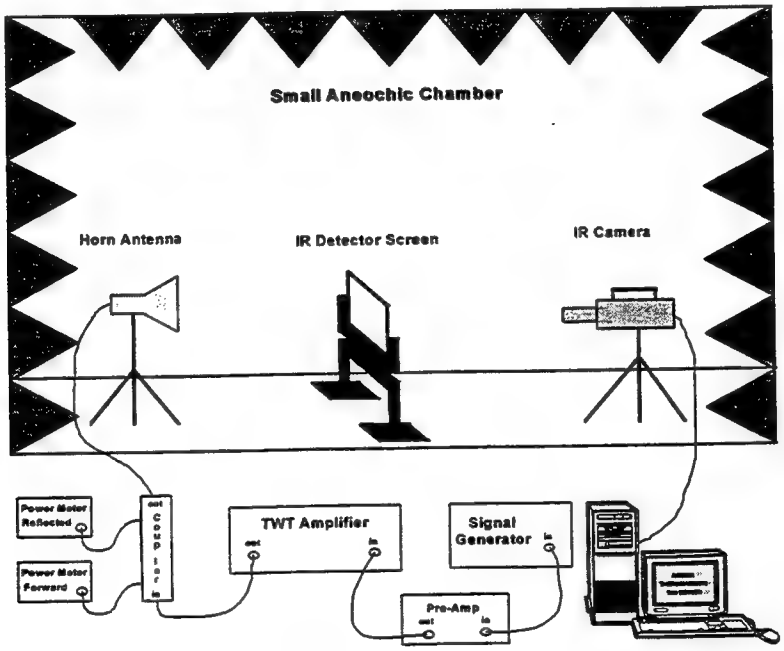


Figure 1 - IR Setup

The EM radiation absorbed by the detector caused the temperature at the surface of the paper to rise above the ambient temperature. The computer screen displayed the

temperature data with the use of different colors corresponding to the various temperatures.

Methodology

At the beginning of an experiment, the ambient temperature was recorded. The signal generator and amplifiers were then turned on and the temperature began to change in proportion to the EM field intensity. The pattern of the standard gain horn (SGH) was concentric circles with the highest temperature in the center. The trials lasted for four minutes. This duration was determined as a result of experimentation. It was long enough for the temperatures to stabilize. Various tests were conducted to pin-point any sources of error. These include changes in the distance between the horn and the screen, the angle of the paper to the horn, and the power level. A strip of data down the center of the IR detector was recorded to see how the temperatures changed over the vertical range of the horn pattern. The differences between the maximum temperature during the trial and the initial maximum temperature were graphed and analyzed. Figure 2 shows the contour plot of measured IR data for a standard gain horn.

Figure 2 - IR contour plot

The method is quite accurate. The plot from the collected data can be compared with a calculated picture. Figure 3 shows the computed magnitude of a standard gain horn. The two pictures are similar and show the same general pattern. They were both taken at 0.5 m.

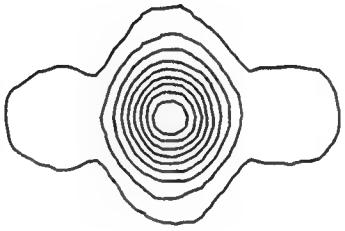


Figure 3 - Computed SGH Reference Magnitude

Many experiments took place at more than one distance from the horn antenna. They were performed at varied distances between 0.25m and 2.0m. Angles between 0° to 45° were tried with a fixed distance of 0.5m, a fixed power level, and a frequency of 4.0GHz. Many other experiments were done with changes in the power level. For one, the power level began at -40dBm and was increased by 1dBm every four minutes. The average maximum temperature was taken at the beginning of the test and at each four minute interval until a 15° increase in temperature was reached.

Results

1. Distance - The IR detector was placed perpendicular to the direction of propagation of the horn antenna. A laser measuring device was used to line up the center of the IR detector with the center of the horn. It was used to line up the center of the IR camera with the center of the horn as well. Many of the same tests were conducted at various distances to view the effects of distance on intensity. One test performed involved taking measurements from 0.5m to 1.0m, in increments of 10cm. Figure 4 shows a vertical strip of data in the center of the horn and how it decreases as distance increases. This allowed the effects of distance on field intensity to be determined.

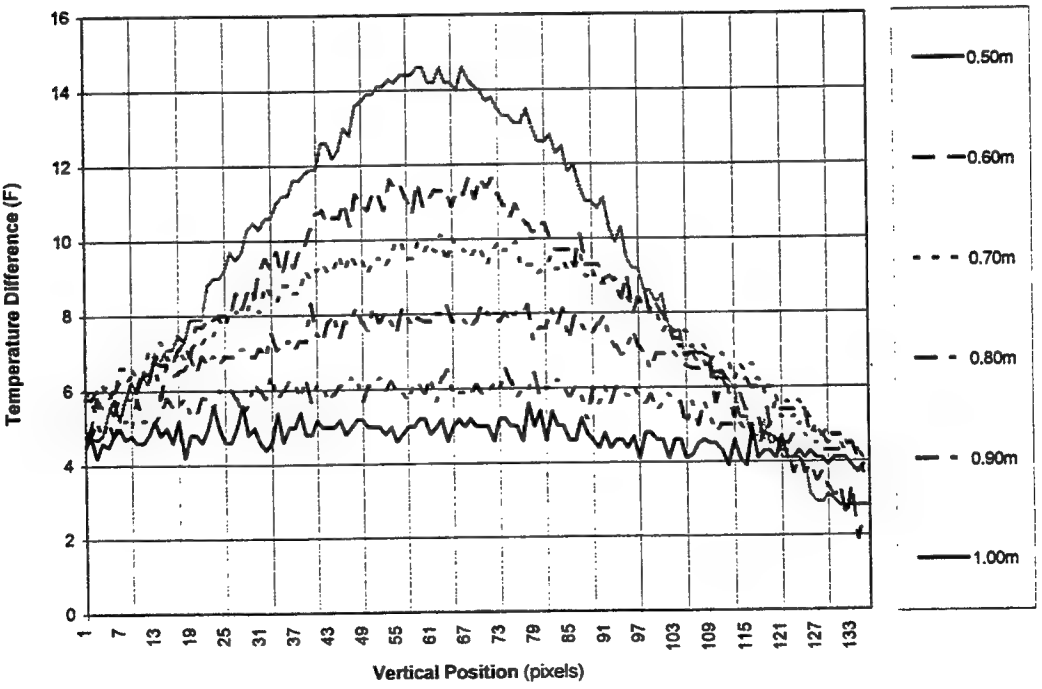


Figure 4 - Distance graph

2. Angle of Incidence - With the distance, power level, and frequency fixed, angles between 0° and 45° were tested in increments of 10° and including 22.5° with respect to the horn antenna. Theoretically, the temperatures at 0°, and at the center of the paper, should be the same as the temperatures at 10°, 20°, and so on at the center of the paper. Results displayed however, that as the angle increased, the temperature change decreased. This was thought to be an effect of the improper alignment of the stand. If the center of the paper was not directly over the pivot point, the same strip would not be sampled every time. It was slightly corrected by a modification in the top disc of the stand. A new disc was made. The center hole was placed very close to the center of the paper. This changed the results slightly but still was not completely accurate.

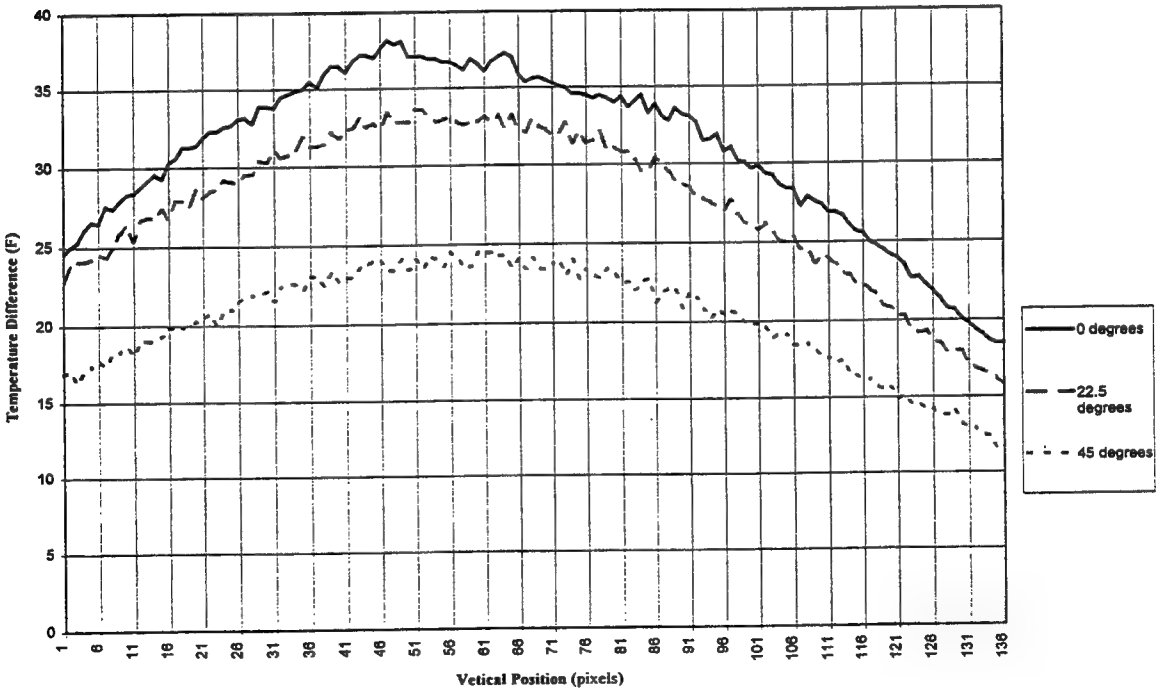


Figure 5 - Angle graph

3. Power - Several experiments were done involving changes in power. They took place in order to determine how much of an effect a small change in power may have on the results. Also to see if an increase in temperature as a result of an increase in power could be predicted. The amount of power put forth by the signal generator effects the results of the experiment. The power is measured in dBm (decibels). By increasing the amount of power generated, the intensity of the radiation that reaches the detector is increased. Thus, the temperature is increased. Also, when the power is increased the forward and reflected powers are increased.

Some experiments were conducted that would create an upward plot as the power was increased. In most cases the power began at -40dBm and was increased by 1dBm every four minutes. The power continued to be increased until a fifteen degree rise above the ambient temperature was reached. The differences between the maximum temperature during the trial and the ambient temperature at the beginning of the trial were plotted. A straight line approximation and a polynomial curve were fit to the plot.

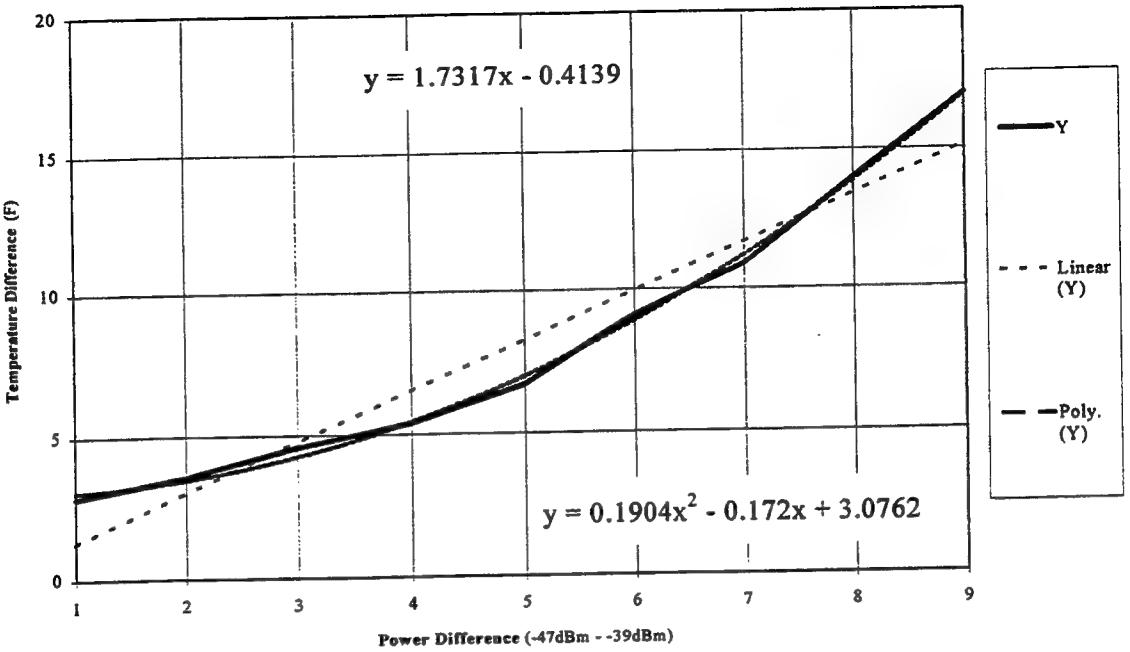


Figure 6a - Line and Curve fit (100ohm paper)

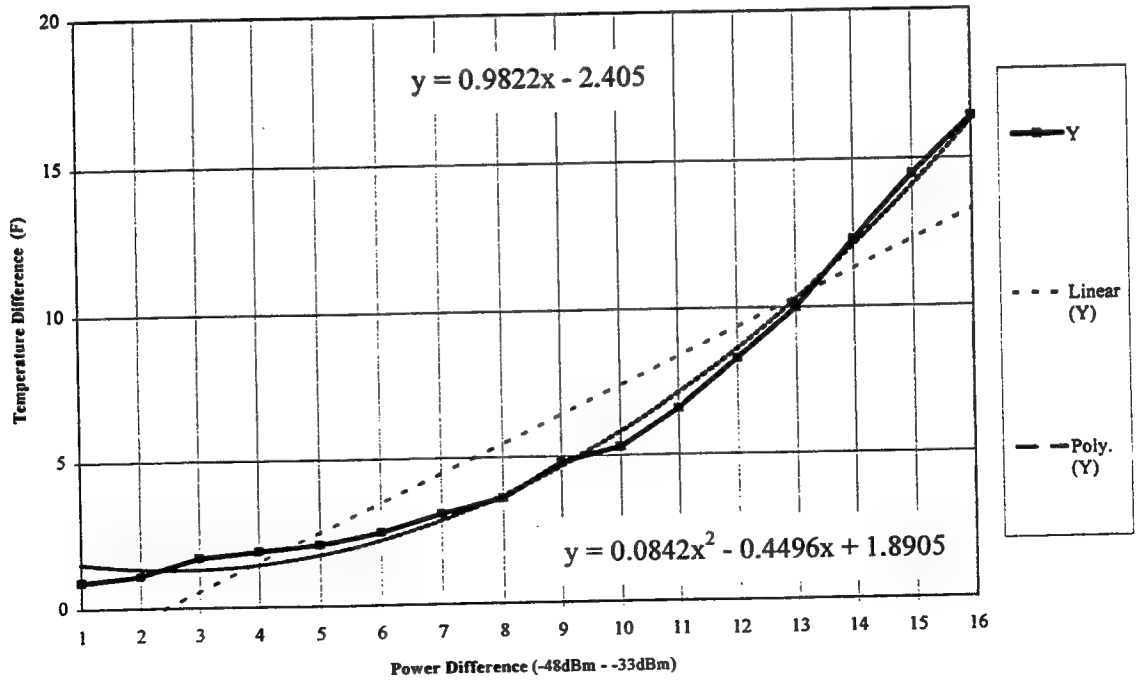


Figure 6b - Line and Curve fit (1500ohm paper)

Conclusion

The patterns from the IR data and from the computed plot are closely related (figures 2 & 3). They both have their highest temperatures in the center and have two side lobes. The computed plot has a scale with more intervals so there are two more rings. The lowest temperature ring on the actual IR plot (figure 2) seems to be flatter than the calculated plot (figure 3). It might be due to a sensitivity of the paper or because of the resolution of the camera.

As the distance between the horn and the detector increased, it was harder to get data because the field intensity lessened. With different angles, positioning is quite critical. If the center of the horn is not exactly lined up with the center of the detector, the results will not be accurate. Also at different angles, the amount of reflected power increases. As the power was increased, the plots became less linear. Smaller increases in power resulted in data that was more linear.

During the experimentation, many margins of error were narrowed while some were eliminated. The measurements are quite accurate and a step closer to being used as a diagnostic technique in the field.

Equipment

Hewlett Packard
Synthesized Signal Generator 83732A
s/n A987996

M/A - Com Microwave Power Devices, Inc.
Solid State Amplifier
Frequency: 2.0 - 4.0 GHz
Power Output: 2 Watts
s/n A991716

Logi Metrics
High Power TWT Amplifier
model A600k
s/n A991721

Bi-directional Coxial Coupler
model 3022
s/n A991719

Hewlett Packard
Horn antenna
model G281A

Hewlett Packard
436A Power M

SPELL CHECKING WITH A DIRECTORY-TRIE IN PROLOG

Matthew A. Lam

Proctor Sr. High School
Hilton Avenue
Utica, NY 13501

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

August 1996

SPELL CHECKING WITH A DIRECTORY-TRIE IN PROLOG

Matthew A. Lam
Proctor Sr. High School

Abstract

A spell checking program using the Amzi! Prolog programing language was constructed, utilizing directory-trie structures to store the database of words. Words are placed in lists under directories and file names corresponding to the letters in each word. When the spelling of a word is checked, the file containing the word (if it exists in the database) is consulted and checked for a match. If there is an error, letters are manipulated in an attempt to find a close match, which would hopefully be the intended word. The final program has an error rate of 1.112%, and a speed of 1.354 words per second. This is considered to be slow, and is the consequence of loading large lists of words into memory.

SPELL CHECKING WITH A DIRECTORY-TRIE IN PROLOG

Matthew A. Lam

Introduction

Spell checking programs are very useful in proof reading documents, and correcting spelling errors. There are numerous programs today that will check text documents, word by word, and search out any grammatical mistakes. New spell checkers are being written and released all the time in an attempt to create the most efficient program possible.

Discussion of Problem

In creating a good spell checker, we need to know what makes such a program better and more efficient than another. The single most important ability a spell checking program needs is accuracy. The program must be able to pick out a misspelled word and then find the intended word, or the closest spelling possible, to replace the mistake. To do this, the common causes for spelling errors need to be identified.

There are four common types of errors that people frequently make when they type. One type of spelling mistake is caused by having a different letter in a word from the one intended. Another common error is for two consecutive letters in a word to be out of order, or switched. Other errors can be caused by missing a letter, or by having an extra letter.

Another important quality for the program to have is speed. We don't want to wait hours just to check the spelling of a five line paragraph. By having a simple straight forward program, the spelling of a word can be referenced very quickly. If a program gets too involved with looping and backtracking, the program will slow down, which should be avoided.

My goal was to write a fairly simple, but efficient spell checker in the Amzi! Prolog programming language. Final results were obtained in about seven weeks, most of which was spent learning to program in Prolog {1}.

Methodology

The basis for the program was taken from an algorithm originally written by McHale and Crowter {2}. It uses Prolog facts to represent words in a database. Each letter of a word is stored as ASCII code in a series arguments. This form of storage quickens the rate at which words are found, by considering only words in the database with the correct arity. This is represented here as the number of letters in each word. In addition, Prolog variables can easily be used to replace or substitute for incorrect or missing letters. When the program runs, it takes a word, changes it to ASCII code, and compares it with the database.

Unfortunately, with a large database of thousands of words, the file needed to be consulted is huge. Besides taking a long time to load, a file containing the entire English dictionary would be so large that it might not fit in memory.

To speed up the program and allow for a large database of words, a directory-trie structure was to be used. In this format, each word is stored by taking the letters from the word, and creating a file system in which each letter is a new directory. Thus, the word "example" would be stored as "c:\e\x\a\m\p\l\e". In the directory representing the last letter of each word, a file would be created, containing the same word converted to ASCII code in a Prolog fact, so that it could be consulted by the main program for the spell checker.

To keep the directory structure down to a reasonable size, and because the system will only allow for directories to go so deep, only the first seven letters were to be used for storage reference. After the seventh letter directory-structure was created, the complete word would be stored in the file. Other words with the same first seven letters would simply reside in the same file. For example, both "associate" and "association" would

be placed in a file such as: "c:\a\s\s\o\c\i\a\word.pro". By swapping the files in temporary memory, the program can access a database of words that would have been too large when combined.

For the purpose of checking the program, a medical dictionary, MeSH {3}, was used as the source for words. The reasoning behind this was that there are many complicated words in a medical dictionary, so if the program worked it would be able to be used with just about any database of words.

Another tactic used to test the program was to create a list of words from the original list, with one of the four common errors made at the end of the word. This helps to simulate a worst case scenario for each word being tested. By placing the errors at the end, the program must cycle through all the rest of the letters before it will find the intended word. All four error types were created for each word, in order to fully test the spell checker.

Results

After writing a program to create the directory-trie structure, I discovered that even when restrained to seven letters, the directories took up too much space in the hard drive's memory. The root directory branches out 26 times, once for each letter. After that, each letter branches out an average of 9 times (consonants usually branch once for each vowel, and vowels branch out many times). After a while, the number of branches would decrease, but even if each branch extended to only one more letter, it would mean that the previous number of final-branch directories is now doubled. As a result, I rewrote the program so that the first two letters are made into directories, and the third letter is used to name a file which substitutes as both the third directory and the "word.pro" file. As an example, the word "example" is now stored as "c:\e\x\a.pro". At this point, the full potential of a directory-trie had been lost, but I felt it was still posible that the program could still run somewhat efficiently.

Originally, the files and directories were stored on a 4 MB RAM drive, which quickly ran out of space.

The final program was setup to get information off both the RAM drive and the hard drive. For some reason, space on the hard drive was used up more quickly than on the RAM drive. While the space used on both drives should have been nearly equivalent, the hard drive used up over ten times more space. I believe this is caused by the difference in size of the allocation units.

Because the words are only divided up three letters deep, there are too many words in some of the files. When a large file is consulted by the spell checking program, it takes longer to load the file than what would be preferred. The delay caused by consulting large files results in an average speed of 1.354 words per second overall for the spell checker. With small files containing five or so words, the program will take less than half a second to check each word. Unfortunately, in files having many words with the same first three letters, the consulting time can be over two seconds per word.

Another drawback of the program is that since only small lists of words are consulted at a time, the program assumes that the first two letters of each word must be correct. In other words, if you spelled "example" as "ezample", the program would not be able to find the intended word. I felt that this was not a major problem, since such a mistake would usually result in a completely different word from the one intended. The setup of the database makes the program run much faster, by consulting somewhat narrowed down lists, rather than huge files. Because the actual system is set up to search three letters deep, if the program can't find any matching word, it consults a different file containing words with the same first two letters, which takes a bit longer to load. Without these two-letter group files, the program would have to assume that the first three letters are always correct, which is not very likely to happen in actual documents. Also, the program cannot be made to only assume that the first letter is correct. If it was written to consult another list containing words with the same first letter, files would be huge, and it would take several minutes to fully load. into memory.

Fortunately, the ability of the spell checker to find the correctly intended word is quite high. After testing each of the four error types on 18,229 different words from the medical dictionary, there was only an error

rate of 1.112% for the program's ability in finding the right word. Most of the errors found were not fixable by any spelling program. For example, if you put a "z" in place of the "c" in "vitamin c", there's not very much any program can do to determine which type of vitamin you were referring to (the program would choose "vitamin a", since it would be alphabetically first). Other unfixable errors included terms with classification numbers that had altered or missing digits.

The most common program errors were made when a word was missing the last letter, in which there was a total of 434 errors. Most often in these cases, the missing letter, along with the program's priority to try to replace existing letters first, led to the formation of a different word. The fewest errors, only 68, occurred when an extra letter was added. The other two error types, having replaced letters or switched letters, tallied up to a total of 194 and 90 errors, respectively.

Conclusion

I think that the actual error rate of the program, when used on a regular English dictionary, would be much lower than what was tested. A lot of the errors that the program made would most likely not appear, or would happen less frequently in an actual document.

The average speed of the program was disappointing. The best way to get around it would be to increase the depth of the directory structure, dividing the words into smaller groups. Unfortunately, that would require too much storage memory, making the program impractical. When stored on a hard drive, the files took up a lot more space than I expected. I think this is due to allocation of files and directories on the drive. If they could be organized more effectively, it would allow enough space to increase the depth, thereby speeding up the program.

Overall, the program does okay, but is actually a little too slow to be practical. However, it could possibly be modified or used as the basis for a more efficient program. A directory-trie structure is a powerful

basis for a search tool, but if the word being searched out is spelled wrong, the program must backtrack through many directories. This could cause a directory-trie structure to be impractical altogether in the use of spell checking, unless the program used an alternative method in which directories can be more easily searched. One possibility would have been to cycle through directories in an attempt to simulate the way a Prolog variable substitutes for letters. I was unable to implement this approach, but more information was examined by Berghel {4}.

References

{1} Clocksin, W. F. and Mellish, C. S., *Programming in Prolog,* Third, Revised and Extended Edition. (1987), Springer-Verlag.

{2} McHale, M..L. and Crowter, J.J., *Spelling Correction for Natural Lanuage Processing Systems*. (1996).

{3} MeSH (1996) Medical Subject Headings. National Library of Medicine. Rockville, MD.

Note: Changes were made to the original list of words. Terms were first placed in Prolog facts, and then each individual letter was changed to ASCII code. Finally, groups of coded words were stored in multiple files.

{4} Berghel, H.L., *A Logical Framework for the Correction of Spelling Errors in Electronic Documents*, Information Processing and Management, Vol. 23, No. 5, (1987) 477-494.

Joanna D. Lisker's report was not available at the time of publication.

Pamela L. McNeil's report was not available at the time of publication.

Using Spreadsheets and Programming in a UNIX Environment

Anthony Perritano

Sauquoit Valley High School
Oneida Street
Sauquoit, NY 13456

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

August 1996

Using Spreadsheets and Programming in a UNIX Environment

Anthony Perritano
Sauquoit High School

Abstract

The Rome Laboratory was conducting the Core Business Audit , this was the examination of all business activities on the Griffriss Air Force base complex. The Core Business Audit included the source of funding, core business activity, directorate used, type of funding and many other headings. Graphs were made representing comparisons and contractions. This was complied using Microsoft Excel version 5.0. The Image Product Archive (IPA) team also under went a revision of their web browser from version 2.3 to 3.0. The help file system was upgraded and reorganized using UNIX systems. Hyper text Markup Language (HTML) programming was used, embedded in C source code. The IPA Interface Control Document was put on-line using HTML programming on UNIX based systems. For the Import Product section of the IPA browser, functions were written in C programming language. These tasks were performed with the collaboration with others. The functions checked the input of the user and return error messages if the input was incorrect. The functions created were a format check and a range check of the input.

Introduction

As computers become more widely used they are performing more complex tasks. Businesses are taking advantage of this, by using spreadsheets and databases to increase productivity and efficiency. Spreadsheets can make graphs, compute mathematical data, sort and filter data, and group it. With spreadsheets a business can keep track of sales, productivity, finances, or just about any thing. Graphs can be made to compare different parts of the business to each other. A graph can give a clear view of all business involvement, such as the Rome Laboratory Core Business Audit. As the software evolves it gets easier to use and more user friendly. Using a spreadsheet and making graphs have become easier to do. As programs evolve, the world wide web and the Internet are growing rapidly. More and more businesses and people are using the Internet in many different ways. For example, businesses can put products and services on the net and people can create personal homepages too. The Image Product Archive (IPA) interface browser is a web browser that can be accessed by the Internet. The IPA browser is a program in which users from the around the world could order and select products. A product can be any thing from the specifications of a jet to picture of New York City. This program is continuously being upgraded and expanded. As the Internet grows so will its influence on daily life and business

Methodology

The information for the Core Business Audit was compiled by many different directorates and branches in the Rome Laboratory. That data was put into a Microsoft Excel spreadsheet. This was done using a Macintosh personal computer. The headings of the spreadsheet were the title of the project, core business activity, directorate/branch, total funding, type of funding, source of the funding, percent of core business involvement, start date and end date. The title represents the name of the program. The Core Business Activity is the specialization of that branch. The core business activities are Dissemination, Exploitation & Manipulation, Protection, Processing, Collection and Storage & Retrieval. Collection can be defined as the arrangement and assemblage of data from sensor systems functioning at frequencies

throughout the electromagnetic spectrum. Processing is the use of computers (both hardware and software) and processes to accept raw sensor data given by Collection components. The employment of information in order to gain the greatest improvement is Exploitation & Manipulation. Dissemination is the precise and steady movement of information to all components in the war fighting force. This provides explicit situation perception. Placing data away to be used later is defined as Storage & Retrieval. Guarding one's own information and data is defined as Protection. These can be done by different branches such as IRDS or C3. The source of funding represented what group financed the program. Funding sources included the Air Force, ARMY and many others.

The data from the core business templates that directorates filled out were inserted into the spreadsheet. The data was put under the appropriate heading. From there, the Excel Autofilter was engaged to list the data by the Core Business Activity. Then the Sort feature was used to group each headings data together and put it in order. Once these procedures were completed, graphs were made. The graphs represented the percentage of different headings in a Core Business Activity. For example, a graph of the percentages of different directorates was made for each Core Business Activity. This format was also used for the funding type, source funding and source percentages. These four graphs were made for each of the four Core Business Activities which totaled sixteen graphs.

The IPA browser's help file system, for its users consisted of HTML files that was extremely unorganized. Some of the HTML files were unreadable and very long. The code size was reduced and the HTML was and cleaned up. Tags were separated and rewritten. The sections of the HTML files were separated and put into there own files. For example, the HTML code that explained a button received its own HTML file. Each button received its own HTML file. This was also done for a variety of different sections. Instead of rewriting the HTML code for a help section and making a long document. The same HTML document was called in conjunction with other HTML documents to form a help section. There where many sections that used the same HTML files and interchanged the other HTML to suit the needs of the that practical help section. It would take a query string and print out the necessary HTML files. This was done using C programming with the HTML embedded inside.

The IPA Interface Control Document (ICD) is a document that provides tables and charts to help clients increase the efficiency of the IPA browser. The ICD document was previously only on paper. The document was scanned into the computer using a GIF format. From there the scanned images were copied on to a UNIX computer. Using the UNIX vi text editor the HTML was programmed. Here is a sample page:

```
<HTML>
<HEAD>

<TITLE>IPA Interface Control Document</TITLE>

</HEAD>
<BODY>

<IMG SRC="/V3/PROPER/docs/ipaICD/gifs/122_6.gif">
<A HREF="/V3/PROPER/docs/ipaICD/TableCon.html"><IMG SRC="icons/back.gif"></A>
<A HREF="/V3/PROPER/docs/ipaICD/define.html"><IMG SRC="icons/next.gif"></A>
<A HREF="/V3/PROPER/docs/ipaICD/TableCon.html"> <IMGSRC="icons/home1.gif"></A>

<HR>
<I>IPA Browser Interface Version 2.3<BR>
</I>
</BODY>
</HTML>
```

The tag <HTML> is used to include the entire HTML document. This tells the browser what format the document is when it is contacted. The </HTML> informs the browser that the HTML document is done. The <BODY> tag contains the main content of the of the HTML document, much like a letter. All the content that is going to be exhibited should be contained in the body tag. The <HEAD> tags contain the general information about the document. The <TITLE> tag is located inside the head tags. The display window is labeled by the title. The title is not part of the document text. The title should be short and give an overview of the document. The image tag or <IMG SRC=**> allows images to be inserted and displayed in a HTML document. The SRC represents the source and following Universal Resource Locator (URL) for transmitting that image. Together the <IMG SRC> tag represents the image source. The image source tag can be integrated with the <A HREF=**> tag. With the integration of these tags it produces a clickable image. The <A HREF=**> provides the image with an

anchor to another HTML page or a different section of that page. The ** represents the URL of the image's designation. The <HR> tag draws a horizontal line that can be used to separate text and other content. The <I> produces italic text within its tags, while the
 inserts a new line.

When a page of the ICD document is loaded, it is displayed as a GIF. Each page is a separate GIF. Under each image three buttons are displayed. One button is linked to the previous page that the user was just on. Another button takes the user to the next page, while the last button takes the user to the table of contents which contains a link to each page. A statement, separated by the horizontal line, in italics displays the current version of the IPA browser.

When the user enters the import product section of the IPA browser they are required to input a certain combination of ASCII characters. If the user entered in the wrong input an error message would pop up. That error message would inform the user that they have done something wrong, but not tell what part of the input was wrong. A program was written in C to inform the user of their actions. One function was written to valid the format of the input in the format column. The other functions checks to make sure the input in the range column is in the given range. The format specifies the format of the ASCII character data. Abbreviations are used, A equals the ASCII upper and lower case characters from A to Z. This also includes periods, blanks, underscores and hyphens. N equals zero through nine ASCII numeric characters. All special ASCII printable characters that are numeric or alphabetic is equal to S. X is equal to all printable ASCII characters. An example of a check is:

```
/*****************************************************************
**        Function to check the format of ASCII character data.        **
*****************************************************************/

#include <stdio.h>
#include <ctype.h>

#define SPACE          32
#define HYPHENS        45
#define PERIODS        46
#define UNDERSCORES 95

int check_format (char *str, int len, int format)
{
 int i
```

```
enum format
{
  A
  N
  S
  X
};


 if (format == A)
 {
   for (i = 0; i< len; i++)
   {
        if( !(isalpha((int)*(str+i)) && (*(str+i)== SPACE ) && (*(str+i)== HYPHENS) && (*(str+i)==
PERIODS) && (*(str+i)== UNDERSCORES) ))
      return (0);
   }
   return(1);
 }
```

Before any function is written, declarations and include statements must be made. Declarations tell the compiler what variables are needed for the program and how much room to allocate. Libraries are collections of functions. For example, stdio.h is a library of functions, this includes printf and scanf functions. Another library used, is ctype.h, which includes functions for checking character types. The type int must be put in front of the variable name to tell the compiler that i is being declared as an integer, which will be used as a counter. Other declarations are define statements such as (SPACE = ACSII character 32) and the declaration of an integer variable named i. The function will be given a pointer to a string, length(len) of the string and format of the strings. This happens in the parenthesis after the name of the function. Inside the function enumerated types are defined for the different formats. Enumerated types provide a tracking method. Enum assigns values to the collection starting at zero. For example, A is equal to zero, N equal to one, S equal to two and X equal to three. If the format is equal to A, then a counter starts. It starts at the beginning of the string and continues until a null character is found. At each position the character is turned into a integer. This is done by (int)*(str+i), which type casts the character at position i of the string(str) to an integer. The if statement states if it is not equal to any of the given values then return a zero, or false. This project is still being worked on and will

down and return a one, or true. Inside the if statement, isalpha checks to see if there is any alpha numeric characters, checks the value of the given integer to the ASCII value define in the beginning of the function. For example, if the character being looked has a value of 32 it is equal to a SPACE. The && symbol is the conditional operator AND, allows to check for multiple characters at once, instead of embedded if statements. N through S are checked in a similar fashion.

Conclusions

All of the projects are still in a working stage. The HTML page is fully functional. The Core Business Audit is still being compiled and updated. The format and range functions are in testing, and the help page of the IPA browser is nearing completion. With the collaboration with others these projects are on the road of successfully being completed.

References

(Judy Kay and Bob Kummerfield, C Programming In A UNIX Environment. (New York: Addison-Wesley Publishing Company, 1989) 3,65.)

Michael A. Scarpulla's report was not available at the time of publication.

HYPERTEXT MARKUP LANGUAGE: CAUGHT IN THE WEB

Patricia M. Swanson

Holland Patent High School
8079 Thompson Road
Holland Patent, NY 13354

Final Report for:
High School Apprenticeship Program
Armstrong Laboratory

Sponcered by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington DC

and

Rome Laboratory
Rome, New York

August 1996

HYPERTEXT MARKUP LANGUAGE: CAUGHT IN THE WEB

Patricia M. Swanson

Abstract

HyperText Markup Language, or HTML, is the computer language of the World Wide Web. The World Wide Web, also known as WWW or W3, was first proposed in 1989 and has grown rapidly in its short time of existence. It has become a significant aspect in the lives of many, and in turn HTML has become more important. Both play a role in the information exchange that exists across the thousands of computers encompassed by the Internet throughout the world. Since it is so widely used by people in all different parts of the world, of all different races, religions, and cultures, with unique purposes, it is made to be easily understood and adaptable.

HYPERTEXT MARKUP LANGUAGE: CAUGHT IN THE WEB

Patricia M. Swanson

Introduction

The World Wide Web was initiated at CERN, the European Laboratory for Particle Physics in Geneva, by Tim Berners-Lee. It allows a vast amount of information to be available to a broad range of people throughout the world. It is based on hypertext, which allows the reader to have some flexibility because he or she does not have to read in a linear fashion, but instead, can jump to different areas of the document or even to other documents to get the information that is needed. Since the WWW represents the information on the Internet graphically, it allows the reader to view graphics, listen to sounds, or see movies as part of the experience.

The World Wide Web software is based on a client-server architecture. A Web client, or Web browser, such as Mosaic or Netscape, is a program that sends requests for documents and presents them on the screen for the user. It requests the documents from a Web server which finds and sends the appropriate documents back. This is all initiated by the user selecting a hyperlink, or link, by clicking with the mouse or using the arrow keys. The document is then presented to the user so that he or she can utilize the desired information.

Problem

To help the public better understand what goes on inside Rome Laboratory, a web site has been placed on the Internet. Throughout this page, a user can link to different branches within the lab. Before these pages can be added to the Internet, they need to be approved by public affairs. The Knowledge Engineering Branch (C3CA) had a web site in progress, but it needed to be updated. A few new pages

needed to be added while others needed to be edited. To do this, the HyperText Markup Language, or HTML, needed to be learned and understood.

Methodology

HyperText Markup Language is the language of the World Wide Web. It was invented, as was the idea of the web, by Tim Berners-Lee. It is a plain text document that uses tags to indicate formatting or structural information. An HTML tag consists of a left angle bracket (<), the specific tag name, and a right angle bracket (>). Tags are usually paired so that they have both a beginning and an end tag. The end tag is identical to the beginning tag except that a slash (/) precedes the tag name. For example, the following are the required elements in an HTML document:

<html> ..... </html>
<head>..... </head>
<title>.....</title>
<body>.....</body>

The previous tags, however, do not need to be in lowercase letters because HTML is not case sensitive. The tags <html>, <HTML>, and <HtmL> are equivalent.

The HTML tag tells the browser that the file contains information in the HTML language. The extension .html must also be used when naming the document to indicate that it is HTML . The HEAD tag identifies the first part of the document including descriptive information such as the title. The document's title is displayed not within the text area, but in the browser window, usually at the top. It is identified with the TITLE tag. The BODY tag is the last tag that is required in an HTML document. It encompasses the largest part of the document containing all of the information displayed within the browser window.

The document that is viewed on the user's screen is created within the BODY tag. The user,

however, cannot control the exact appearance of the document. He or she can mark a section of the document as a paragraph, heading, title, etc., but cannot be sure how each browser will interpret the markup. For example, a paragraph may be indented by one browser while another may only leave a blank line between that paragraph and the last. A document containing the same information is presented differently in various browsers due to their different abilities. HTML was designed to tell the browser "what" the information is, not "how" to display it. When the information is labeled as what it is, the browser will determine how it should look and display it so that it looks acceptable.

After the HTML start tag, the HEAD tag and its corresponding elements, and finally the BODY start tag are in place, the displayed document actually begins. Usually a heading begins the document. There are six levels of headings ranging from one to six. The first heading in the document should be tagged <h1> and the headings following should be tagged <h2>, <h3>,...through <h6>. Heading one is the largest and/or most bold of the six headings. Headings should not be skipped, such as going from <h1> to <h3>, to get a desired visual effect. The heading text should be followed with the heading end tag. Many times the <h1> text is the same as the title because the title is not displayed in the browser window.

The next tag that is commonly used throughout the body of the document is the paragraph tag. A paragraph is labeled using <p>. The paragraph end tag can be omitted because browsers can understand that a new <p> tag means that the previous paragraph has ended. When aligning paragraphs, which will be explained later, the paragraph end tag is necessary. Using a paragraph end tag cannot hurt. Therefore, if the user is unsure, he or she should use it just to be on the safe side.

Lists are another important aspect in the body of an HTML document. There are three types of lists: unnumbered, numbered, and definition. The unnumbered list, also called a bulleted list, and the numbered list are very similar. The main difference between them is that the items in a bulleted list are marked by small dots while a numbered, or ordered list, marks the items with numbers. An unnumbered

list would begin with the <UL> tag followed by each list item labeled <LI> and finally end with the< /UL > closing tag. A closing tag for the <LI> tag is not necessary. The numbered list follows the same pattern, but the <UL> and </UL> tags are replaced with <OL> and </OL>.

The third and final type of list is the definition, or description list. It creates a list of items and a description for each of the items. The definition list begins with the <DL> tag and ends with its corresponding </DL> tag. The tag <DT> identifies the description's title, or the word that is going to be defined. It does not need an end tag. The actual description does not need an end tag either. It is labeled with a <DD> tag.

All of the previously mentioned lists can be nested. This means that one or more lists are contained in another. These nested lists do not all have to be the same type of list, but instead a nested list can contain all or some combination of the three different types of lists.

Linking is an important aspect of HTML since the language is based on hypertext. It allows for access to documents containing important information in conjunction with the current document that the user is viewing. The first step in creating a link is to start the anchor. This is a left angle bracket (<) with the letter A following it and then a space. Next the document being linked to is specified using HREF="URL" and the right angle bracket (>). Finally, the text that will appear highlighted or underlined in the document is typed and the anchor is closed using < /a >. Therefore a link would look something like this when coded in HTML:

<A HREF="the linked document's URL">highlighted text

Since certain symbols have special meaning in HTML, they cannot simply be used as is in the text of the document. Instead an escape sequence is used. Escape sequences are used for the left and right angle brackets and the ampersand as well as other characters not available in the plain character set such as the tilde. The escape sequences are case sensitive unlike the rest of HTML however. Upper and

lowercase letters cannot be interchanged. Here are some escape sequences:

&lt	the escape sequence for <
&gt	the escape sequence for >
&amp	the escape sequence for &
&ntilde	the escape sequence for an n with a tilde (~)
&Egrave	the escape sequence for an E with a grave accent (')

Although the user cannot control the exact appearance of the document, there are tags that allow him or her to create certain effects within the document. These tags let the user create words that are in italics or are bold as well as create horizontal rules and background graphics and colors. There are many stylistic tags some of which are shown below.

Stylistic HTML Tags:

< i >...< /i >	creates italic type
< b >...< /b >	creates bold type
< code >...< /code >	creates a fixed-width font
< strong >...< /strong >	creates strong emphasis
< cite >...< /cite >	for titles of books, films, etc.
< em >...< /em >	creates emphasis
< tt >...< /tt >	creates typewriter text
< pre >...< /pre >	creates a fixed-width font
< br >	forces a line break
< body background=" URL " >	creates a background graphic
< body bgcolor = " color code " >	creates background color
< body text = " color code " >	changes text color
< body link = " color code " >	changes link color
< hr >	creates horizontal rule
size = #	changes thickness of rule (change #)
width = %	percentage of window covered by rule

* Both size and width are enclosed in the horizontal rule, hr, tag

On the following pages there is an example of an HTML document. The first page shows a sample document as it would appear on the users screen. The next page shows a link to a different page from the example page. It would be what would appear if the user selected the link labeled "here". Following these

two pages are two pages which show the HTML coding for the sample page. These pages give examples of previously discussed topics and also show other tags such as the table and address tags and how to align within a tag which have not yet been discussed.

An Example of What You Can Do:

Here's an example page. This is a short centered paragraph under heading one. Isn't it simple?

This paragraph is here to tell you about line breaks. All the empty space between this paragraph and heading two was done with line breaks. Notice that some spacing is automatic though, like in between paragraphs and headings.

Lists

- Here is an example of a **numbered** list:

    1. Apples
    2. Oranges
    3. Peaches

- Here is an example of an **unnumbered** list:

    - Red
    - Green
    - Blue

- Here is an example of a **definition** list:

HTML
: Hypertext Markup Language

WWW
: World Wide Web

Below is a link to a table. This table gives information about tables. If you were to click on the word **here** or select it using your arrow keys, you would be brought to a new page. It is found in *A Beginner's Guide to HTML* which can be found on the Internet at http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html. The following page (12-10) shows the table as it would appear in the browser window.

Here is information about tables.

This concludes my sample page.
You may contact me at
swanson@somewhere
(just kidding).

Table Elements

Element	Description
<TABLE> ... </TABLE>	defines a table in HTML. If the BORDER attribute is present, your browser displays the table with a border.
<CAPTION> ... </CAPTION>	defines the caption for the title of the table. The default position of the title is centered at the top of the table. The attribute ALIGN=BOTTOM can be used to position the caption below the table. **NOTE:** Any kind of markup tag can be used in the caption.
<TR> ... </TR>	specifies a table row within a table. You may define default attributes for the entire row: ALIGN (LEFT, CENTER, RIGHT) and/or VALIGN (TOP, MIDDLE, BOTTOM). See Table Attributes at the end of this table for more information.
<TH> ... </TH>	defines a table header cell. By default the text in this cell is bold and centered. Table header cells may contain other attributes to determine the characteristics of the cell and/or its contents. See Table Attributes at the end of this table for more information.
<TD> ... </TD>	defines a table data cell. By default the text in this cell is aligned left and centered vertically. Table data cells may contain other attributes to determine the characteristics of the cell and/or its contents. See Table Attributes at the end of this table for more information.

Table Attributes

NOTE: Attributes defined within <TH> ... </TH> or <TD> ... </TD> cells override the default alignment set in a <TR> ... </TR>.

Attribute	Description
• ALIGN (LEFT, CENTER, RIGHT)	• Horizontal alignment of a cell.
• VALIGN (TOP, MIDDLE, BOTTOM)	• Vertical alignment of a cell.
• COLSPAN=*n*	• The number (*n*) of columns a cell spans.
• ROWSPAN=*n*	• The number (*n*) of rows a cell spans.
• NOWRAP	• Turn off word wrapping within a cell.

```
<html>

<head>
<title>A SIMPLE EXAMPLE</title>
</head>

<body>

<h1>An Example of What You Can Do:</h1>

<p align=center>Here's an example page.  This is a short centered paragraph under heading one.
Isn't it simple?</p>

This paragraph is here to tell you about line breaks.  All the empty space between this paragraph
and heading two was done with line breaks.  Notice that some spacing is automatic though, like in
between paragraphs and headings.<p>
<hr size=4 width=75%>
<br>
<br>

<h2>Lists</h2>
<ul>
<li>Here is an example of a <b>numbered</b> list:<p>
<ol>
<li>Apples
<li>Oranges
<li>Peaches
</ol>
<br>
<li>Here is an example of an <b>unnumbered</b> list:<p>
<ul>
<li>Red
<li>Green
<li>Blue
</ul>
<br>
```

Here is an example of a definition list:
<dl>
<dt>HTML
<dd>Hypertext Markup Language
<dt>WWW
<dd>World Wide Web
</dl>

Below is a link to a table. This table gives information about tables. If you were to click on the word here or select it using your arrow keys, you would be brought to a new page. It is found in <cite>A Beginner's Guide to HTML</cite> which can be found on the Internet at http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html. The following page (12-#) shows the table as it would appear in the browser window.

<A HREF="http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html#TA">Here</A> is information about tables.

This concludes my sample page.
You may contact me at <address>swanson@somewhere</address> (just kidding).

Results

The HTML language was used to enhance the Knowledge Engineering Branch's (C3CA) web site. Once pages were added and others were edited, it was sent to the public relations department. After it is approved, it will be available on the World Wide Web through Rome Laboratory's web site at http://www.rl.af.mil:8001/.

Conclusion

In conclusion, the HyperText Markup Language (HTML) is a simple language designed to structure information for use on the World Wide Web. Even with little or no programming knowledge, HTML is easy enough so that it can be used world wide. Whether it be for personal or business use, it is a useful tool for getting "connected" to the information superhighway.

References

"About the World Wide Web and the Web Community"

http://www.w3.org/pub/WWW/History.html

"The Bare Bones Guide to HTML"

http://coba.shsu.edu/help/internet/barebone_table.html

"The Beginner's Guide to HTML"

http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

"The Beginner's Guide to HTML"

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/html-primer.html

"Colors"

http://www.infi.net/wwwimages/colorindex.html

"Composing Good HTML"

http://www.cs.cmu.edu/~tilt/cgh/

"Entering the World-Wide Web: A Guide to Cyberspace"

http://www.szote.u-szeged.hu/mirror/entering.www/guide.toc.html

"Getting Started on the World Wide Web"

http://www.law.washington.edu/webstart/webstart.html

"HTML BRIEF"

http://www.msn.fullfeed.com/~hadron/htmlbref.htm

"HTML Quick Reference"

http://www.cc.ukans.edu/info/HTML_quick.html

"The Hypertext Markup Language (HTML)"

http://weber.u.washington.edu/~rells/RSSO/html.html

"Information on the World Wide Web"

http://sunsite.ust.hk/homepage/w3info.html

"Introduction to HTML"

http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/htmlindex.html

"A Little History of the World Wide Web"

http://www.w3.org/pub/WWW/History.html

A Study of the Computer Networking Environment

Brian B. Tuch

New Hartford High School
33 Oxford. Rd.
New Hartford, NY 13413

Final Report For:
High School Apprenticeship Program

Sponsored By:
Rome Laboratory of Griffiss Air Force Base

August 1996

A Study of the Computer Networking Environment

Brian B. Tuch
New Hartford High School

Abstract

The problems and intricacies of a computer network were studied. In order to study these problems a computer to be used as a print sever, World-Wide-Web or WWW server, and a remote dial-in server was set up. The setup of this computer included installation of Linux and all other applications which would serve to make the computer an integral part of the network. Most problems encountered along the way which required in-depth attention involved integration between UNIX and Macintosh hardware.

A Study of the Computer Networking Environment

Brian B. Tuch

Introduction

As the computer becomes an ever increasing necessity in the office and home environments so with it grows the need for a connection or link among computers. Never in the world's history has the idea of a "global community" been so popular as it is today. In order to provide such an office or global community a network or physical connection between computers must be established. In this scenario the physical link had already been set up, but this alone did not allow for the computers to communicate. The software behind the communication had to be established in order for the computer to act as a print, World Wide Web, and dial-in server.

Discussion of Problem

The problem here was to implement the new computer as a print, World Wide Web, and dial-in server. Previously there had been a computer for print and dial-in serving, but this new computer was an upgrade from the old one. This new computer was also to be set up as a way to exchange data between linux users; this is where the use of a WWW server becomes important. The computer was also intended to be used during off hours as a means of dialing in remotely and having access to the Lab's network and to the internet, therefore it was necessary to set up the dial-in server. The final problem was to enable the computer's ability to act as a print server, and more specifically allow it to be a gateway for printing to the printers connected by LocalTalk.

Methodology

The first step in setting up such a server is to install an Operating System. The apparent question is which

Operating System, which was quickly answered with Linux. Linux is a UNIX-like, free Operating System which comes with its own source code and is designed to be used in a network environment; it was the perfect choice for all the above reasons.

The next step was to set up a new print server and add greater capabilities than the previous one; this involved allowing the printers to print multiple file format types including GIF, JPEG, and TIFF files. In Linux print setup is done by adding printer entries to a printcap file. The printcap file holds information on filters, printer names, and spool directories. The task at hand was to enable eleven remote printers for use by UNIX computers. The problem was that nine of the eleven printers were connected by LocalTalk, a Macintosh device, and AppleTalk, a Macintosh protocol. A program called NetaTalk was installed because it allowed the UNIX computers to see and communicate with the nine AppleTalk printers through the use of a Fastpath. The Fastpath is a hardware device which interprets TCP/IP signals over the Ethernet and is able to convert these signals to the AppleTalk protocol for use over the LocalTalk. By using NetaTalk and the Fastpath a UNIX computer is able to print on an AppleTalk connected printer. As for the other two printers which were connected to the network via Ethernet and the UNIX machines there was no setup necessary because UNIX was inherently able to see these printers without additional software. An extra step taken in setting up the print server was to add a filter package called APSfilter which is intended for use in printing files other than Postscript. APSfilter allows the user to print out GIF, JPEG, BMP, and DVI files to a normal postscript printer.

After setting up the computer as a print server the next step was to enable its ability to act as a WWW server. The computer was to be set up as a local electronic information booth for Linux users to exchange information. The software that was installed for web-serving was Apache. After compiling and installing Apache some simple configuration was needed; this involved editing various config files which contained information on who could access the web site, what directories the web pages were in, and other intricacies

of web serving and web pages. Of particular interest in the configuration files was the specification of who could access the web site; because this web site was meant only for use within the lab its access was limited to those within the Rome Lab network domain. The final step in setting up the web server was creating a home page which displayed that the site's primary purpose is to assist those who have Linux problems.

The final task in this investigation of networks was to configure the Linux system to allow for remote PPP, Point-to-Point Protocol, logins over a telephone line using two modems. The first step in doing this was to configure the modem correctly using its AT commands and then saving them as a profile. The next step is to set up the inittab file which will listen to the modem serial ports for a ring, if and when the ring occurs it will pass it on to getty_ps. With this done the next step was to set up getty_ps which allows users to log in over the phone and subsequently locks the modem ports to tag that they are in use. Uugetty, part of the getty_ps package, accomplishes the file locking and sets the speed at which the modem is to connect. There is also another file, named after the modem port, which sets the times at which the phone lines are to be answered. This file was implemented with non-working hours so that the phones could be used as voice devices during work hours. The final files which needed altering were the diphosts and passwd files which were changed to acknowledge those users who would be using the dial-in system.

Results

For the most part, all methodology behind this investigation was successful in implementing the computer as a print, WWW, and PPP dial-in server. However, some problems were encountered along the way. One such problem involved the printers. In order to get the printers working for use by the UNIX systems it was necessary to change the filters and in many cases use more than one filter. When APSfilter was added it was incapable of handling multiple printers and therefore a variable had to be added which specified to which printer the output, after filtering, is to be sent. Another problem which still needs to be fixed is the

PPP dial-in which does not always answer the phone as it should. On a good day the PPP dial-in will work beautifully however on some days it will not even attempt to connect. This problem remains a mystery to be solved.

Conclusions

Over the past eight weeks I have gained a great deal of knowledge and experience in the areas of Linux and networking. I have learned that setting up and maintaining a server on an office network is not an easy task and that many mind-challenging problems can be found along the way.

References

PPP-HOWTO

Serial-HOWTO

Gilly, Daniel. UNIX in a Nutshell. O'Reilly & Associates. USA. 1994.

Hahn, Harley. A Student's Guide to UNIX. McGraw-Hill Inc. New York. 1993

"What is Linux ?" Linux Journal. May 1996: pg 4.

DETERMINING THE STATIC VOLTAGE DISTRIBUTION ON CIRCUIT STRUCTURES

Cheryl Zaglaniczny

Whitesboro High School
Rte. 291
Marcy, NY 13403

Final Report for:
High School Apprentice Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

August 1996

DETERMINING THE STATIC VOLTAGE DISTRIBUTION ON CIRCUIT STRUCTURES

Cheryl Zaglaniczny
Whitesboro High School
Rte. 291
Marcy, NY 13403

Abstract

The static voltage distribution on circuit geometries was studied. The static voltage distribution in the vicinity of conducting structures is described by Laplace's equation. Numerous techniques are available for solving Laplace's equation. For this work, an iterative averaging scheme was chosen. Numerical simulation issues of accuracy, grid resolution and boundary conditions were investigated.

DETERMINING THE STATIC VOLTAGE DISTRIBUTION ON CIRCUIT STRUCTURES

Cheryl Zaglaniczny

Introduction

The final goal of the project is to determine the time domain electromagnetic response of circuit geometries due to high frequency pulses propagation. The first step in this procedure is to determine the static field pattern attributed to the power and ground distribution system throughout the circuit. The static field pattern will essentially provide the initial conditions for the complete time domain analysis. Therefore I investigated the static voltage distribution on a variety of structures. The static voltage distribution is described by Laplace's equation. One way to solve Laplace's equation is to use an iterative averaging scheme described in the next section.

Methodology

Laplace's equation, given in Equation 1, describes the static voltage distribution, V, in the vicinity of a conductor structure. To approximate the second order derivatives represented by the Laplacian operator, we can use the averaging scheme shown in Equation 2. In Equation 2, V(i,j) represents the voltage to be computed. The four values for V on the right hand side of the equation represent the voltage at the cells above, below, and to the left and right of the current cell. This averaging procedure involves the following steps. First, divide the modeled region into individual cells. Then, set the bordering cells to the prescribed initial conditions. Next, assign an initial guess to the interior cells. Finally, apply the iterative averaging technique to calculate the voltage value for each cell.

$$\nabla^2 V = 0 \tag{1}$$

$$V(i,j) = \frac{V(i+1,j)+V(i-1,j)+V(i,j+1)+V(i,j-1)}{4} \tag{2}$$

A computer program was written to apply that formula to every cell top to bottom, left to right and record the data. When it has gone through every cell once, it has completed one "pass." It then iterates or repeats another pass- again applying the formula. After obtaining a certain level of accuracy the program will stop iterating, and record the end results of that last pass. For example, when the maximum difference between the voltage at the same cell on successive passes is less than the desired percentage change, the program will stop. All of this results presented in the next section are representative of two-dimensional structures (assumed to be infinite in the "Z" direction).

Problem Description and Results

Trough

The first problem is represented by Figure 1 and is called a trough. It is a relatively simple distribution problem and was primarily used to test the techniques used in solving a static voltage distribution problem. The cover plate has a fixed voltage of 100 volts while the ground and side plates are assigned a value of zero volts. This case was investigated for three resolutions (grids of 10x10, 50x50, and 100x100). All three gave similar results. The computed voltage distribution for the 100x100 grid are seen in Figure 2. In this figure, the electric potential distribution is shown using a grey-scale, black representing zero volts and white representing the highest voltage throughout the grid.

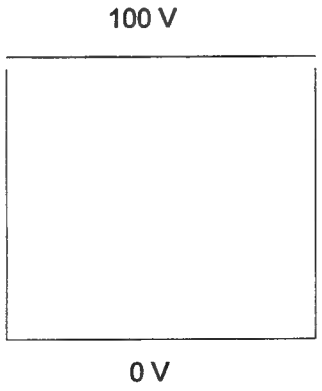


Figure 1 Two-dimensional trough geometry

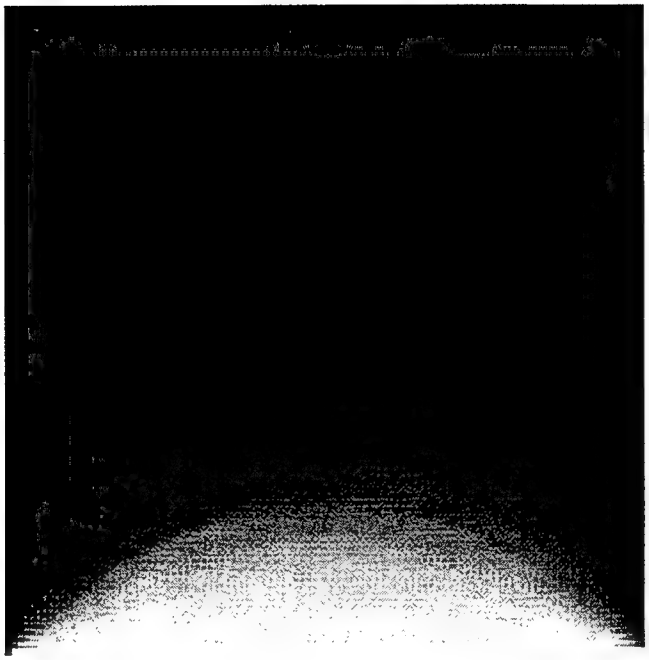


Figure 2 Computed Voltage distribution for 2-D trough

An important consideration for this type of computer solution is grid coarseness. The advantage of a fine grid is that you can obtain more detailed results. However, to get those results, it takes many iterations which can become very time consuming. Coarse grids give you an idea of what the final results would be like and take much less time to compute. So before you choose your grid tolerance you should first consider how precise you want your results and how much time you can afford to spend on obtaining them. It may also depend on the complexity of the problem.

This first case was relatively simple because it was a totally enclosed structure (we were interested only in the voltage distribution on the interior of the structure). A more complicated case is that in the next section, the stripline. The stripline is an open geometry which forces us to consider treatment of the boundary where the computational domain is terminated.

Stripline: Case 1.

In this case, we measured the static voltage distribution on a two-dimensional circuit which contains a stripline in the center. The cover and ground plates were set to a voltage of +5.0 and -5.0. The stripline and the side plates both had a value of zero volts. The structure is shown in Figure 3. Ideally, we would not want to have any side boundary because electromagnetic theory tells us that the true results have fringing effects at the edges of the plates. The results for this case are shown in Figure 4. In an attempt to conquer this obstacle, I changed the boundary conditions as described in Case 2.

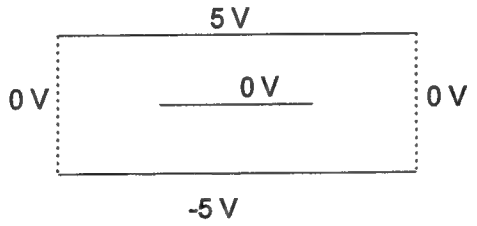


Figure 3 Case 1 Stripline configuration with boundary conditions

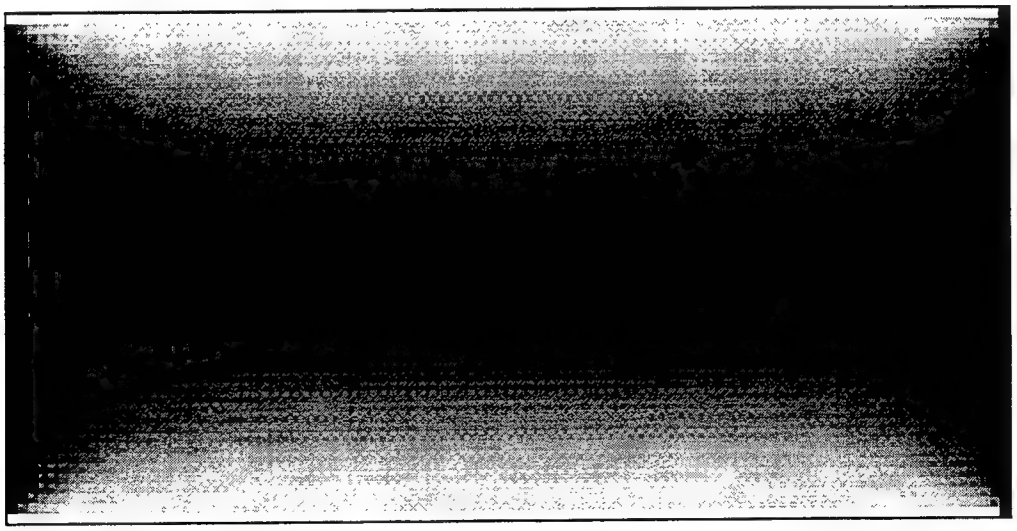


Figure 4 Case 1 - computed voltage distribution for boundary conditions of zero volts.

Stripline: Case 2.

In Case 2, shown in Figure 5, the side plates are assigned a voltage varying linearly from +5.0 volts to -5.0 volts. The results are shown in Figure 6. However, these proved to be unrealistic results for the case we're interested in because it again neglected the fringing fields. The boundary conditions effectively simulate plates of infinite length and by doing so this fixed the voltage distribution inside the structure to be linear variations. The next step to determine the effects of the fringing is described in Case 3.

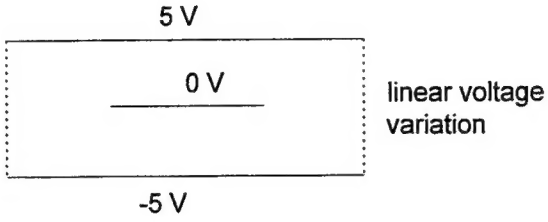


Figure 5 Case 2 Stripline configuration with boundary conditions

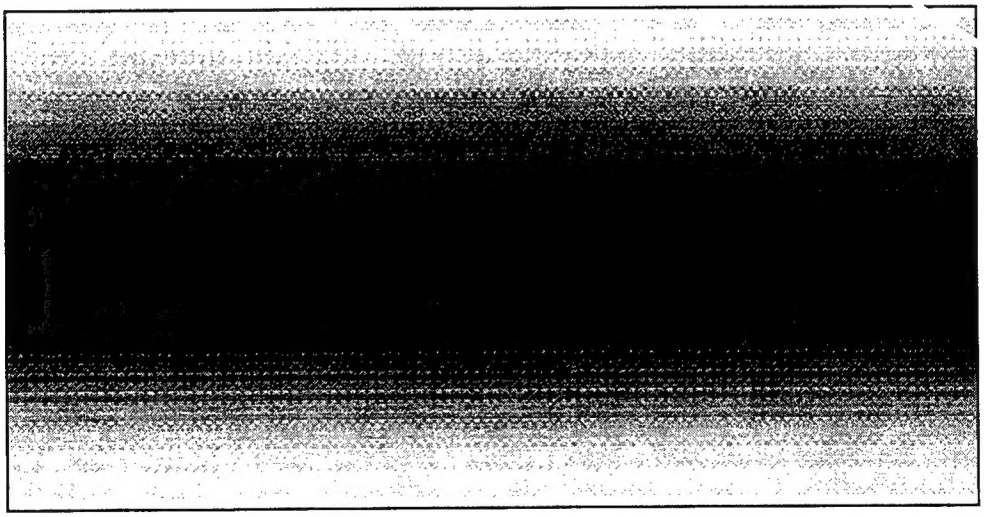


Figure 6 Case 2 computed voltage distribution

Stripline: Case 3.

A more realistic model of this stripline case is presented in Figure 7. In this model we embedded the stripline geometry into a ten times larger computational space and assigned the boundaries to a voltage of zero volts. This is justified because electromagnetic theory says that the voltage should drop to zero far away from the source of voltage. Ideally, we would embed the stripline geometry into a computational space of infinite extent. However, such a case is impossible to solve with a computer. The program would not fit onto any computer and would need to compute forever.

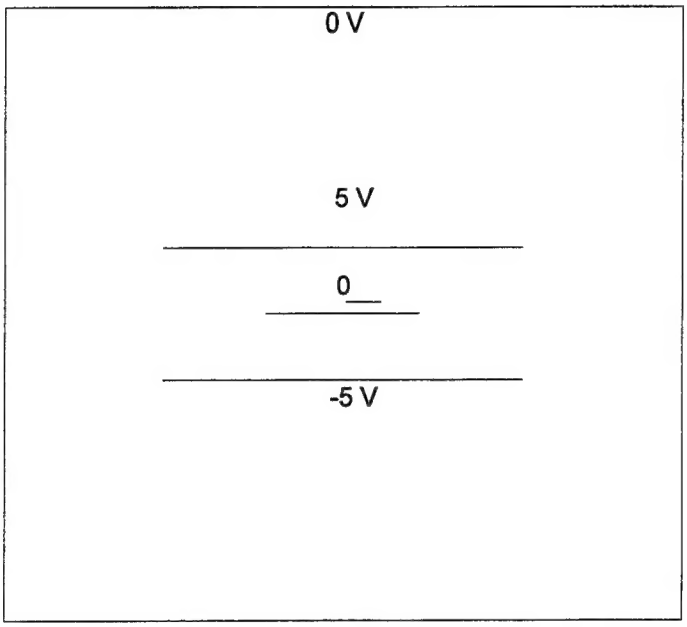


Figure 7 Case 3 - stripline embedded in large computational domain.

The technique we used then allows you to observe the more realistic flow of energy around and at the ends of the oppositely charged metal plates. The computer can now solve the structure without going on forever because we have set boundaries to approximate the actual situation. The results are shown in Figure 8.

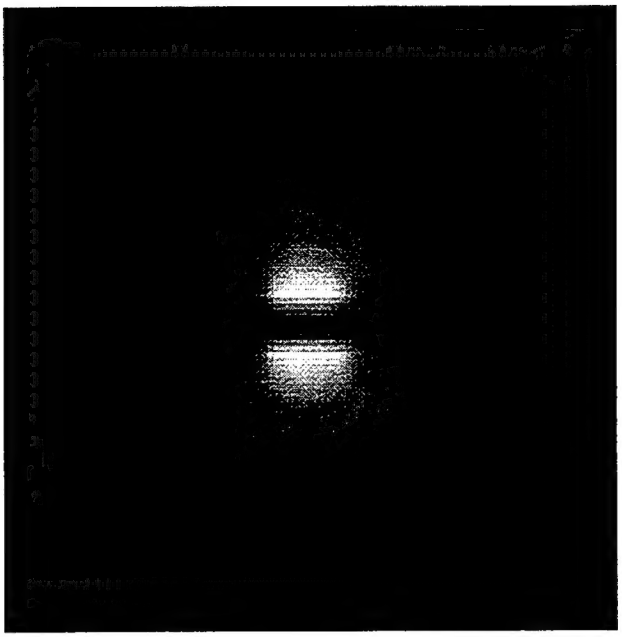


Figure 8 Case 3 - computed voltage distribution for embedded strip line.

Conclusions

There are two conclusions that can be drawn from our work. First, we investigated the advantages of different grid resolutions. Second, the boundary condition is critical to accurate numerical results in the interior of the grid. The current state of research is a search for an effective boundary condition to use when solving Laplace's equation for open circuit structures. Ideally, we want to find boundary conditions which do not have an effect on the interior voltage distribution. Such a method is still being researched.

Addendum

When one thinks of a laboratory, they generally envision people in white coats taking measurements, beakers bubbling, and lasers bouncing from here to there, but a laboratory is much more than that. There are many different steps you must take before you can enter that final stage of experimentation. One of these steps involves computer simulations.

While at Rome Laboratory, I worked in the Electromagnetics and Reliability Directorate. I studied how computer simulations could be used to solve electromagnetic problems. However, before I could start the actual programming of the simulation, I first had to learn how to "speak the computer's language." The language that I learned was FORTRAN. After becoming accustomed to FORTRAN, I used a workstation with the UNIX operating system and the "vi" editor to construct a program which would give me the results I needed to solve the problem given to me.

The programming process isn't always easy. Sometimes your syntax (computer language) is not correct or the program is not giving you what you expected it to. In these cases you must "debug" the program. This consists of a wide range of tests and corrections to insure that your program is doing what you expect it to (solving the desired problem). The debugging process can consist of correcting syntax errors, typos, and errors in logic. However, just because everything seems to be working and producing answers, does not mean that the answers the program is producing are the ones you want. You have to analyze the data and investigate to see if the solution is, in fact, the true answer that you were looking for.

Computer simulations, such as the three I worked with, are essential to any experimental process. They allow you to construct artificial environments and test the effects of different electromagnetic structures and conditions in those environments without having to leave your desk. When you are satisfied with the results of your program and they reflect what you thought would take place, you can then put them to use in reality. I would like to become a medical doctor so mastering this step-by-step process of analyzing and logically evaluating data is a valuable experience and will help me in my future